Information Flow Control with Decentralized Labeling Model in Information Security

Veli Hakkoymaz and Cigdem Bakir*

Yildiz Technical University, Computer Engineering Department, Istanbul, Turkey
E-mail: veli@ce.yildiz.edu.tr; cigdem.bakir@erzincan.edu.tr
**Corresponding Author*

Received 26 May 2020; Accepted 11 August 2020;
Publication 16 December 2020

Abstract

Data security aims to prevent the use, modification, and spread of data by unauthorized people. In this study, our purpose was to provide data privacy and confidentiality with information flow control in distributed databases. In particular, a decentralized label model was developed that maintained confidentiality, including privacy, with data flow control. This model consists of an actor, an object, and a label. The owners of the objects are actors, and they need to share their data objects with others. Actors label the data objects and then send them out. A label contains the policy statements of data security issued by each of the owners. Each owner sets its own security and privacy policy independently of the other owners. The confidentiality of data in unsecured transport channels is ensured for all the actors in the system by means of labels while the data are in flow. Data objects are spread and shared securely among actors within unsecured environments. In addition, with the path compression, the long node chain that is formed while the data objects are passing between the source node and the destination is broken, so that the objects are retrieved fast, and the cost of access is reduced. This result was shown experimentally by modeling the distributed environment.

Keywords: Label model, data confidentiality, path compression, distributed databases, data privacy.

Journal of Web Engineering, Vol. 19_7–8, 903–930.
doi: 10.13052/jwe1540-9589.19781

1 Introduction

Data security in distributed database systems requires the prevention of the unauthorized use, modification, and propagation of the data. With the rapid development of technology, data security has become an important problem in many areas such as banking, health, e-commerce, and communication. In order to solve these problems expressed as information leakage, the seizure of information by unauthorized persons, the change of information, and the failure to provide information confidentiality, measures such as Information Flow Control (IFC) and Access Control (AC) are used [1].

Data confidentiality refers to the use of data only by authorized principals and the fact that these actors can perform such operations as reading and writing on the data. There are some common points between personal data protection (privacy) and data confidentiality [2]. However, privacy and confidentiality are not the same concepts. This topic has been discussed in many studies in the literature. Privacy is a more complex concept than confidentiality, and humans are a part of its focus. In contrast, confidentiality is a working area of cryptology, which is used in communication security and focuses on data and where encryption methods are used [3]. In other words, the protection of privacy is to prevent personal or corporate, sensitive, and important data from getting into the hands of individuals who might abuse it [4].

The financial data of organizations, personal information related to the diagnosis and treatment process of patients in hospitals, credit card information about customers in banks, and product design information in industry are examples of sensitive and confidential data [5]. These types of information must be protected at the source, the inter-individual circulation, and the target [6].

For example, let us consider a scenario of keeping patient records in two medical centers (Figure 1): The purpose is to share records quickly and securely in both the centers. For updating data at a center, we need the same patient record at both the centers. At each medical center, in addition to the general information such as a patient's name, surname, TC identification number, birth date, birthplace, blood group, gender, telephone, and address, there is also health-related private information that should be kept confidential, such as diagnosis, treatment process, past medical findings, used drugs, laboratory reports, radiology reports, surgeries, chronic diseases, infectious diseases, and pregnancy status. In addition to general and private information about patients, there is also the registration data of the medical center. This

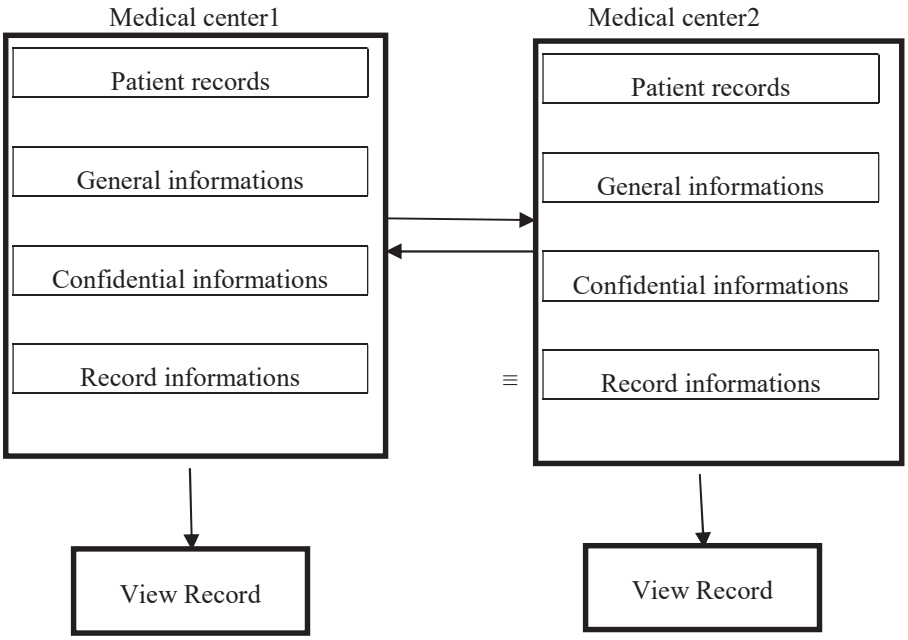


Figure 1 Sharing of information between two medical centers.

example illustrates a common knowledge object and the actors contributing to it. The major actors can be seen as the patient, the doctor, and the medical center personnel. That is, we can assume that the mentioned actors have this information object in common. To be able to solve problems such as missing information about patients and updating the treatment process, each center makes changes in its records. When a center wants to access information about a patient in the other center, it receives and sends the information that it wants to access according to the local security policies. With the information flow control, the realization of these policies is ensured. To be able to ensure privacy, patient records should be shown to the authorized personnel at both the centers, and others should be blocked.

The purpose of this study was to develop a method that allows different users to access the data in a distributed environment and that protects confidentiality. We aimed to investigate methods preventing unauthorized access to data that are accessed jointly by multiple actors.

Data leak, which is also called data breach, is a condition that occurs when the information flow control is not performed. This may have some

undesirable consequences, from a violation of the personal data protection law to the endangerment of national security [7]. Unauthorized, unexpected, and unintentional access and the leaking of these accessed data to third parties may result in embarrassment, the loss of the sense of trust in the institution, or the initiation of legal proceedings against the institution [8, 9].

For example, patient data and the data in the tax forms must be accessed by many actors for data processing or for calculations. Access refers to operations such as reading and writing. The solution of a specific problem, such as the protection of data at the same time while being ensured this access, is an important scientific contribution. Therefore, this study aimed to be the first step in this direction.

Privacy is the most important element of human rights [10]. In particular, patient privacy is very important. Patient privacy includes the diagnosis and treatments that patients want to hide from all people. The fact that the records of these health data are easily accessible by unrelated persons causes a violation of the patient rights [11]. These data may be spread, sold to other people, or used as blackmail material. Therefore, it is necessary to protect patient privacy and to secure patient rights [12].

As a specific example, in medical centers, it is important to determine to whom and with which authorizations should the health data be given and to specify their access privileges. The operations to access the patients' health data for both internal staff and external authorized/unauthorized users are subject to supervision. However, a failure to fully carry out this supervision results in problems such as access and the use of these data by unauthorized third parties or dissemination of data [13]. With the labeling model in this study, the confidentiality of the patient records will be ensured.

In this study, a distributed label model was developed. The scientific contribution of this model is that while the data available to use by the stakeholders are used only by authorized actors easily, it does not allow the use of these data by unauthorized third actors. At the same time, this model contributes to the research of methods that enable the use of jointly used resources without causing information leakage. It is also necessary to reduce the time and calculation cost of accessing the data. With the path compression method, the data are accessed faster and the cost of access is reduced.

Therefore, in this paper, we describe a distributed label model that can maintain data confidentiality with information flow control in distributed databases.

The difference between this study and the other studies on this subject is that with the label model, it targets data confidentiality in non-reliable

actors and environments. Through the labels given to the data, each actor can determine his/her own security policy independently from the other actors and authorize the ones that he/she chooses from the other actors.

In the rest of this article, the related studies, definitions, distributed label model and the path compression method are presented; findings and results are discussed; and recommendations for future studies are made.

2 Related Works

A distributed computing environment can be modeled with a graph data structure. A graph is a data structure consisting of a set of nodes and edges connecting these nodes to one another [14]. If G signifies a graph, its definition is as follows:

$$\mathrm{G} = (\mathrm{V}, \mathrm{E}) \tag{1}$$

$$\mathrm{V(G)} = \{\mathrm{v1},\ \mathrm{v2},\ \ldots,\ \mathrm{vN}\} \tag{2}$$

$$\mathrm{E(G)} = \{\mathrm{e1},\ \mathrm{e2},\ \ldots,\ \mathrm{eM}\} \tag{3}$$

Hence,

$$\mathrm{E} \subseteq \mathrm{VxV} \tag{4}$$

The distributed environment is performed with three types of nodes: storage, worker, and dissemination [15]. The edge indicates the transition of a data object from one node to another. The *storage node* stores objects permanently. The *dissemination node* allows an object to be copied when the object is requested from itself. It checks whether the worker and the dissemination node have the authorization to import objects (with the privacy policy). The *worker node* runs programs. The dissemination node stores the frequently used objects in groups [16].

Each program (software) takes some security measures against the installation of unsafe codes. For example, Java allows the downloading of codes from remote sites. This may cause some confidential information to be transferred to these sites. Users can calculate their own data by installing applets (code snippets) in Java [17]. The code snippets allow the Java code to disseminate over the Internet and run in a browser. However, when the user downloads them, they may access various hidden files and may cause this information to be transferred to various sites. This information needs to be protected from the code snippets coming from these types of users. Java uses the "sandbox" security model against these risks. This security model

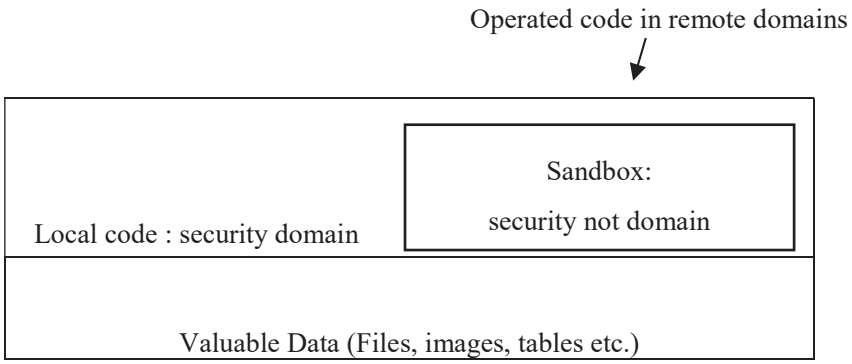


Figure 2 Java security model.

is shown in Figure 2 This model is more reliable in the running of the local codes. Local applications prevent the sharing of the data to untrusted users. However, it is a security model that is used in limited areas, as it causes data leakage in the running of insecure code snippets that are run in large applications and remote areas [18].

Four basic control mechanisms have been proposed to ensure data security in databases. These are access control, inference control, flow control, and data encryption. This study is related to flow control for data security [19].

Access control is performed in three different ways: discretionary access control (DAC), mandatory access control (MAC), and role-based access control (RBAC) [20]. DAC refers to the definition made by the data owner as "grant access to data" or "remove this right." Because it performs optional authorization, it is a rigid security model that is not flexible in the form of "has authority" or "does not exist" [21]. In contrast, the MAC uses the sensitivity level to provide access to data. The user divides the data and the objects into various security levels that include a pyramidal cluster. The information flow is provided according to these security levels (i.e., top secret, secret, classified, or unclassified). Rather than being used in the industry and in companies based on central and mandatory access levels, MAC is a security model for military and intelligence applications for the State [22]. RBAC defines the roles for users according to their tasks and responsibilities in the enterprises. Users use the authorization of the roles assigned to them. Access processes are performed according to these roles. All the users with the same role perform all the operations that this role has [23].

The access control is insufficient to solve problems that are covered by the information flow control because these techniques represent only the authority of the actor [24]. Preventing data objects from getting into the hands of unauthorized persons requires the bringing under control of transport channels that allow them to be forwarded to unauthorized persons. To prevent the violations of confidentiality, the sharing of data objects securely and the performing of functions and calculations are required even though there are insecure nodes in the distributed environment [25]. Data confidentiality can be also performed by encryption, a low-level security method. Encryption is the conversion of data into a different format with a key. After the data reach a recipient from a sender, the recipient converts the encrypted text into plain text [26]. For example, public and private keys are used for the Rivest–Shamir–Adleman (RSA) encryption method. The public key is known to everyone. However, the encryption technique, which is a low-level security technique, poses a key distribution problem in distributed environments. The distribution of the key (password) to users is a security problem by itself [27, 28].

In this study, the label was used as a key, and the data access was controlled through labels. Instead of a central model, a distributed security model in which the security requirements of all the actors were defined was developed [29]. In this model, data obtained from multiple sources were considered the common data of some of the actors in the system. These data could be released just after taking the joint consent of the data owners. While other techniques enable the information flow in secure environments in which there are secure users and secure objects, the label model also checks the information flow in untrusted environments with untrusted actors. Each user in this model determines his/her own security policy independently of the others.

The topic of reviewing the access authorizations is one of the laborious studies under the title of information security [30]. The studies related to this subject are studies having a high cost in terms of time and labor force within the institutions. Especially within complex database structures, a review of the access authorizations cannot always be done cost-effectively; therefore, it may be neglected or cannot be performed in a sufficiently qualified manner [31].

In our study, access control and authorization are ensured in accordance with the actors' wishes, without causing data leakage and with the supervision of information flow control. The actors are able to create their own security, confidentiality and integrity policies in a practical and flexible way.

In addition, when they are done, they can easily change or completely delete these policies. This is also performed at the working time in our study and in both a static and dynamic way. In multi-object environments, where many non-secure actors get access, it has been intended for the actors to protect their data easily by the security policies that they identify themselves. The difference of this study from other studies is that it provides data confidentiality, data integrity, and data consistency together. In addition, rapid access to data has been ensured by the path compression algorithm, which we have proved its effectiveness by also experimental study.

Schultz et al have developed a platform that ensures automatically tracking the data access of the users. Because a user logs into the system separately for each transaction, the authorization control is performed again. The user is forced to perform authorization control at every stage. Data confidentiality is breached if he/she does not perform control only at one stage. This situation creates the need for the authority to be able to be tracked automatically [32]. In our study, however, there is no need for a separate control for both giving and revoking the authority. In our study, there is no need for separate authorization or access control for each transaction such as reading, writing, updating, deleting. In addition, by following the access of malicious actors to the data, it has been tried to prevent information disclosure.

In earlier studies in the literature [20, 30], a separate label was used for each transaction (read, write) performed on the object, and only read and write transactions were carried out. On the other hand, in the study proposed by us, all transactions performed on the object (read, write, update, delete) are carried out using a single label. Thus, by looking at a single label, it is shown with which transaction and how there is an authorization and access control between the actors. In addition to data privacy; we used the path compression algorithm to increase data access speed and reduca data cost. We showed the advantages and success of our proposed alogirthm with experimental study. Since we used statistical approaches in distributed environment simulation, our study has been more successful than other studies [33, 34]. Moreover, unlike previous studies [35–37], data confidentiality and data consistency were realized simultaneously.

3 Definition

In this section, some of the definitions used in this study are presented [38]:

Definition 1: *Access control* indicates who can access the data and what actions they can perform on these data.

Definition 2: *Data flow control* is the control of confidential channels to prevent the seizure of information by unauthorized persons. It prevents information leakage by blocking the transport channels that allow the data to be transmitted to unauthorized persons.

Description 3: *Declassification (downgrading)* is the reduction of data from the upper security level to the lower security level. Transferring data to a lower security level reduces the confidentiality level.

Description 4: *The label* consists of a set of security policies. Each security policy is an expression indicating a partner of the data owners and to whom this partner will give the utilization permit for the data (e.g., as a reader).

Description 5: *Re-labeling* is performed to make changes on the security policies of the label, provided that the safety level of the label is not reduced.

Definition 6: *Privacy* is the phenomenon indicating by whom, when, and by how many users the data of the institutions and the users will be used. It is the restriction of access to data according to the permissions of the users.

Description 7: *Multi-level security level* refers to the definition of the different pyramidal levels for the users and the information objects and the assignment of these users and objects to these levels. Figure 3 shows four different security levels. While top secret refers to the highest security level, unclassified refers to the lowest security level.

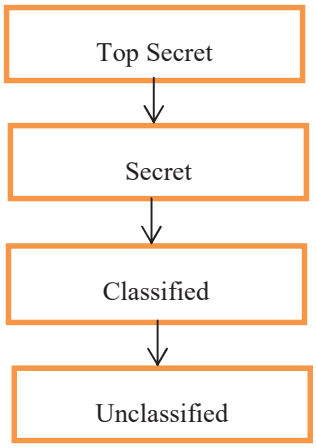


Figure 3 Security levels.

4 Decentralized Label Model

The distributed label model consists of different principals, objects, and labels [39].

4.1 Principal

The term "principal" refers to the data owners and users or user groups performing operations such as giving and receiving authority over the data. The label consists of a list of security policies that are given by the actors. Each actor labels his/her data for data privacy. That is, a label which is conjugated with a data object is determined. In addition, each actor has the right to safely change these security policies separately. This model was developed for unreliable actors and environments. Each actor changes his/her own policy independently from each other and performs re-labeling. To ensure safe re-labeling, all the actors are required to label their security policies with a "reliable" action [40, 41].

In the decentralized label model, the principals consist of groups or roles that change the data that they possess. Principals allow other principals or principal groups to read their own data. This licensing process is shown in the principal hierarchy. Figure 4 shows a sample principal hierarchy. In this figure, X and Y are the representatives of a worker group. Worker Z has two tasks and duties as an engineer and a unit head. In the principal hierarchy, the process of granting authority is transitive. For instance, X→Y should stand for granting authority by X to the Y principal. If X→Y and Y→Z, then X→Z.

4.2 Label

A label is a collection of policies that are created for the protection of the data.

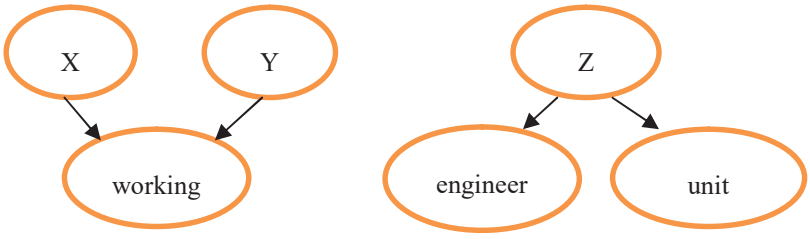


Figure 4 Examples of the principal hierarchy.

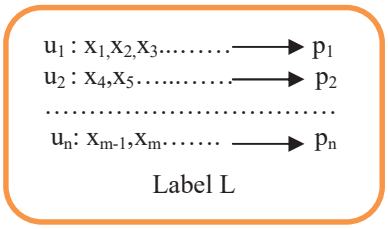


Figure 5 Label L example for the data object.

Figure 5 shows the contents of a label. Here, while $u_1, u_2, \ldots, u_n$ show the owners of the data object from the actors in the system; $x_1, x_2, \ldots, x_m$ refer to the actors to whom authorization is given for the any transaction by the data owners; $p_1, p_2, \ldots, p_n$, i.e., each content definition on the L label, show the security policy of the relevant actor regarding these common data. Each actor who owns a data object determines his/her own policy on the label. Then, one of the actors sends these data object to the other actors with its label.

4.3 Graph Modeling of Labels

A label can be shown with a graph. Let us assume that the label specified for the G graph is L_G.

L_G consists of two parts as {owner:readers:writers:updater:wiper}. The shapes of the arrows in the graph show types the authority to access the data. Here, while "owner" denotes the actors who own the labeled object, "readers" refer to the actors to whom authorization is given for the read transaction by the data owners, "writers" refer to the actors to whom authorization is given for the write transaction by the data owners; "updater" refer to the actors to whom authorization is given for update transaction by the data owners; "wiper" refer to the actors to whom authorization is given for delete transaction by the data owners. The label shown in Figure 6 with the graph G can be expressed in the L_G typing format as follows:

$$L_G=\{\ v_1{:}v_2{,}v_4;\ v_2{:}v_3,\ v_4;\ v_3{:}v_4,\ v_5;\ v_4{:}v_5,\ v_5\ \}$$

The semicolon used when creating a label separates the policies from one another. Accordingly, the L_G label has five policies: $\{v_1{:}v_2{,}v_4\}$, $\{v_2{:}v_3,\ v_4\}$, $\{v_3{:}v_4,\ v_5\}$, $\{\ v_4{:}\ v_5\}$, and $\{v_5{:}\ \}$. While v_1, v_2, v_3, and v_4 denote the owners of the data object to which the L_G tag belongs, v_2, v_3, v_4, and v_5 represent the actors authorized by the data owners for various transactions (read, write, update, delete) on the object.

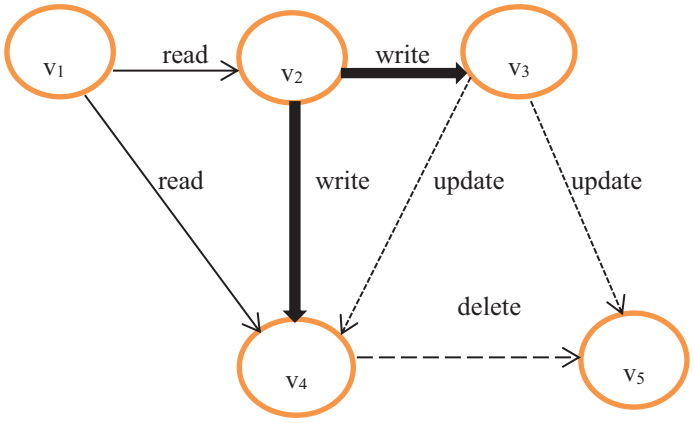


Figure 6 A graph G modeling of the label.

Let us assume that the first policy shows the read operation on the object:

The first policy is expressed with the $v_1 \rightarrow v_1$, $v_1 \rightarrow v_2$ and $v_1 \rightarrow v_4$ edges. This means that the v_1 actor allows the v_1, v_2, and v_4 actors to read his/her data.

Let us assume that the second policy shows the write operation on the object:

The second policy is expressed with the $v_2 \rightarrow v_2$, $v_2 \rightarrow v_3$, and $v_2 \rightarrow v_4$ edges. This means that the v_2 actor allows the v_2, v_3, and v_4 actors to write his/her data.

Let us assume that the third policy shows the update operation on the object:

The third policy is expressed with the $v_3 \rightarrow v_3$, $v_3 \rightarrow v_4$, $v_3 \rightarrow v_5$ edges. This means that the v_3 actor allows the v_3, v_4, and v_5 actors to read his/her data.

Let us assume that the fourth policy shows the delete operation on the object:

It is expressed by $v_4 \rightarrow v_4$, $v_4 \rightarrow v_5$ edges. This means that the v_4 actor allows the v_4 and v_5 actors to delete his/her data.

The last policy is expressed with the $v_4 \rightarrow v_4$, $v_4 \rightarrow v_5$ edges. This means that the v_4 actor allows v_4 and v_5 actors to delete his/her data.

The last policy;

The last policy is expressed with the $v_5 \rightarrow v_5$ edge. This means that v_5 does not allow anyone other than himself/herself to perform any transaction his/her data.

4.4 Data Transfer betweens Principals

The transfer of a data object shows from the u_i actor to the u_j actor. In the meantime, in order for the uj actor to receive and read the data coming from the u_i actor, the u_j actor must be the owner of a policy in L, which is the label of these data, or be included in all the reader lists. This is expressed in the following reading condition:

Condition for reading:

Let $i \neq j$; then, the condition for the u_j principal to read the L-labeled data can be stated as follows:

if $\{1 \leq i \leq n; \forall_i\ u_j \in reader_i[L]\}$ or $\{1 \leq i \leq n; \exists_i\ u_j \in owner_i[L]\}$ {
    u_j has the permission to read the data w/ label L
}
else {
    u_j has no permission to the data w/ label L
}

Whether a received data object can be read or not will be checked with this reading condition. If the reading condition is not met, u_j cannot read this data object. It will have only mediated in the transferring of the data object from one end to the other.

4.5 Relabeling by Restriction

Each data object has a label. When a new value is assigned to an object, the value that this object receives must also be shown on the label. The new label value is determined by placing new restrictions on all the policies on the old label. The new label value of the object must be at least as restrictive as the old label value. This is the rule for re-labeling with a restriction.

The meaning of the $L_1 \subseteq L_2$ expression is that the policies in the L_1 label must be equal to the policies in the L_2 label or the L_2 label may also include other policies in addition to the policies in the L_1 label. In short, the L_2 label contains at least as many restrictions as L_1 or more.

J is the policy of the L_1 label, while **K** is the policy of the L_2 label; hence, for transmission from L_1 to L_2 via re-labeling, the required equation is as follows:

$$L_1 \subseteq L_2 \rightarrow \text{owner}(K) = \text{owner}(J), \text{readers}(K) \subseteq \text{readers}(J) \quad (5)$$

The examples of re-labeling via a restriction are as follows:
L_1 (old label) $\subseteq L_2$ (new label)

Example 1: $\{X{:}Y,Z\} \subseteq \{X{:}Y\}$. While the Y and Z actors are the readers of the L_1 label, the L_2 label has removed the reader Z and allowed only the Y actors to see the data (re-labeling by removing a reader).

Example 2: $\{X{:}Y\} \subseteq \{X{:}, Z{:}T\}$. The L_2 label has removed the Y reader and added a new policy of $\{Z{:}T\}$ to increase the restriction (removing a reader and adding a policy).

Example 3: $\{X{:}Y,Z\} \subseteq \{X{:}Y;X{:}Z\}$. L_1 and L_2 contain equal restrictions.

The data owners on the label control the spread of data through various restrictive processes, such as deleting their own policies or removing a reader. The purpose of this re-labeling is to perform a secure declassification. If the reader that the data owner adds is also added by other data owners, the data that are re-labeled in this format are read by this reader.

5 Our Solution

The data are expressed as objects, and various names are given to these objects. An object is represented by oid, which indicates the identity of this object. The oid consists of the host that indicates where the object is stored, and parts showing which object is on the host (that is, the object number).

5.1 Path Compression

The *path* is a graph structure that shows the transmission of the data objects by the actors (nodes). The nodes are connected to each other in the form of chains. As an object moves between the actors, the object identifier (oid) that shows the identity of the object is updated to include the next address of the object. For example, in Figure 7, an object (oid_1) is transferred to the storage nodes A_1, A_2, A_3, A_4, and A_5, respectively. When the object is transferred, only the address information remains on the previous node and the object is transported to the new node. When the object is moved from the A_4 storage node, it is transferred to node A_5 and the new address is saved as $oid_1(4)$ to node A_4. In other nodes where the object is passed, the address (reference) of the object remains ($oid_1(1)$–$oid_1(2)$–$oid_1(3)$–$oid_1(4)$). The object shown with $oid_1(5)$ is actually located in node A_5. $Oid_1(5)$ in A_5 shows the new oid value of the object. Because object movements create long chains between nodes in this format, the cost of accessing the object increases. In order to prevent long

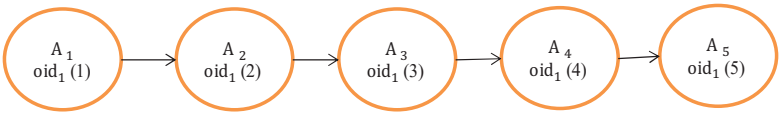


Figure 7 Node chain formation.

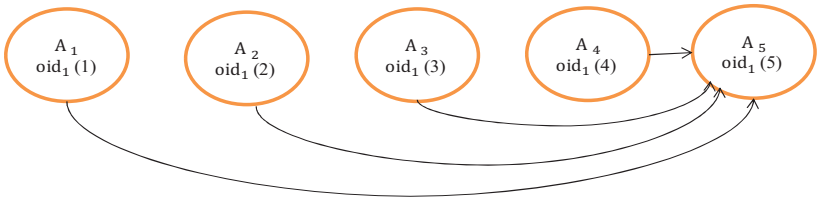


Figure 8 The new condition that occurs as a result of the path compression.

chains, the *path compression method* is used, which gives the result shown in Figure 8. *Path compression* is the process of updating the reference in each node on the path, which starts from the root node to the node where the object is currently located, with the current location address (See Algorithm 1).

5.2 Object Access Process Example

If we desire to locate an object without using path compression, it needs to look at all the nodes that the object passes until the desired node is reached. In Figure 7, when the position of an object in Node A_5 is asked to Node A_1, nodes A_1, A_2, A_3, A_4, and A_5, where the object address is stored, must be accessed, respectively. However, with the path compression method shown in Figure 8, it goes directly to node A_5 (A1 gives the object address in A5 directly). Thus, fast access to the object is possible, and the cost of access is reduced.

The path compression algorithm (Algorithm 1: Algorithm Pathcompress) is an efficient algorithm with linear runtime. This can be easily proven as follows:

Hypothesis: Algorithm 1's operation time is O(n).

Proof: Once the steps of the path compression algorithm are analyzed, the operation time T(n) can be expressed as follows:

$$\mathrm{T(n)} = 2\mathrm{n} + c \tag{6}$$

In this formula,

Algorithm 1 Path Compression Algorithm

```
    Algorithm YolKısalt (Start:Node)
    //Start node
1:  X ←Start;
2:  Y ←Start;
3:  if (X=null veya next[X]==null) return;
4:  // determine previous node to probe (Y)
5:  while (next[X]!=null) do
6:        Y←X;
7:        X←next[X];
8:  end while
9:  // update display
10: Z←Start;
11: while (Z!=X) do
12:       next[Z]←next[Y];
13:       Z←next[Z];
14: end while
15: return;
```

T(n): working time
n: number of nodes in the chain
c: any given fixed number

Because with steps 1–4 ($\mathbf{c}_1$), steps 5–8 (**n**), steps 9–10 ($\mathbf{c}_2$), steps 11–14 (**n**), and step 15 ($\mathbf{c}_3$) are represented and

$$\mathrm{c} = \mathrm{c1} + \mathrm{c2} + \mathrm{c3} \tag{7}$$

equality 8 is obtained. When this function is solved, equality 9 is obtained.

$$\mathrm{T(n)} = \mathrm{Q(n)} \tag{8}$$

5.3 Distributed Environment Simulation

In a distributed environment, some operations related to objects are performed. For example, objects can be accessed or moved. In our study, events related to objects (object-access and object-move) were created as independent events, and a distributed environment simulation was performed. The purpose of this study was to show the effectiveness and the benefits of the path compression algorithm. In the distributed computing environment, let us assume that there are n nodes and k objects. For example, let these n nodes be denoted as d[1], d[2], ..., d[n]. Each node has a local object table. Let k objects be denoted as o_1, o_2, ..., o_k. The object table on each node will

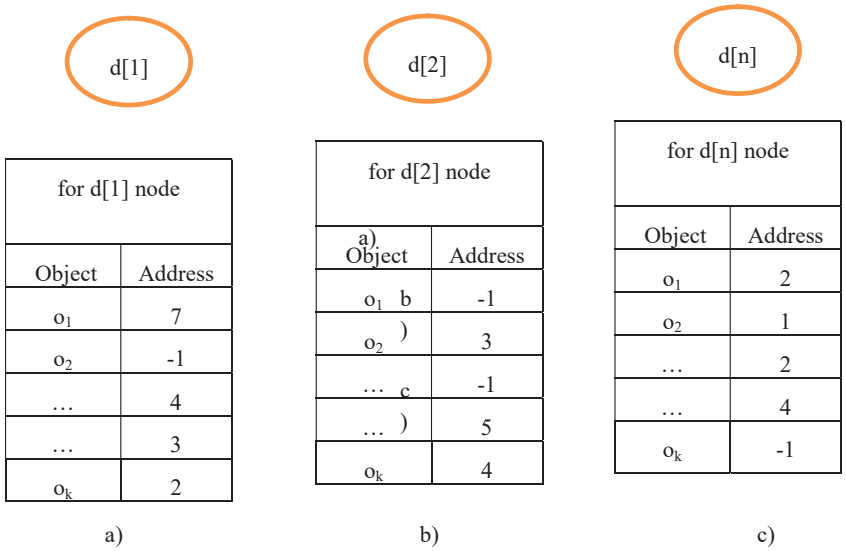


Figure 9 (a) d[1] local object table, (b) d[2] local object table, (c) d[n] local object table.

contain the object information showing that it resides on this node or contains the address of the object if it is in another node. For each node, there is a notation similar to that shown in Figure 9.

Initially, objects are randomly assigned to nodes. This is accomplished with the function F: oi $\rightarrow$ d [j], for $1 \leq i \leq k$ (random (1, n)). Each o_i object is assigned to any d[j] node. We used five functions related to the objects:

Function 1: object_access (i:object, j:node): This function looks at the local object table of node **j** for object **i** to be accessed. If the address shown by this node is -1, it means that the object is located on this node. If not, the node address is retrieved from the **j** node object table and is assigned to **j**. This process continues until the object is found. This function returns the chain length.

Function 2: object_move (i:object, j:source node, x:destinion node): With this function, object **i** is moved from the **j** node, in which it is currently located, to the **x** node. The node where all objects reside is stored in the general object table.

The global object table stored in distributed systems is shown in table. This global object table shows on which node each existing object is. It is always up to date. Furthermore, the object_access(i, j) and object_move(i, j, x) functions operate independently of each other.

Function 3: break_chain (i:object, j:source node, x:destinion node): The length of the chain of the **i** object accessed by the object_access(i, j) function is checked. The length of the chain is equal to the number of nodes that are used to access object **i**. A threshold value **T** is determined. If the length of the chain is equal to or greater than the threshold value, the **i** object breaks the chain starting from the **j** node to the **x** node where the object resides. If the chain length is less than the threshold value, this function is not executed. The appropriate values are determined by changing the threshold value.

Function 4: Book_keeping (i:object, L:chain length): The length of chain **L** is calculated for each **i** object to be accessed. In this simulation, by running the DriverForObjAccess( ) and DriverForObjMove( ) functions at certain rates, we calculated the average and the maximum chain lengths. By changing the threshold value, we re-calculated the maximum and the average chain lengths.

Function 5: compute_ statistics ( ): It records the length of the chain of the object, which is as much as **Zmax**, being accessed. When objects with a length equal to **Zmax** are accessed, the mean chain length is calculated by assuming that sufficient statistics are collected. The mean length of the chain is determined by taking the ratio of the total length of all the objects accessed to the number of objects accessed. The mean chain length is calculated as follows:

$$\text{Mean chain length} = \sum_{\text{i=1}}^{\text{Zmax}} \text{L[i]/Zmax} \tag{9}$$

6 Experimental Study and Results

In our study, distributed environment simulation was performed in two ways as binomial and normal distribution.

6.1 Binomial Distribution

In the distributed environment, some operations related to objects are performed. For example, objects can be accessed or moved. In our study, events related to objects (object-access and object-move) were created as independent events, and a distributed environment simulation was performed. The purpose of this study was to show the effectiveness and the benefits of the path compression algorithm. In our study, access to the objects was shown

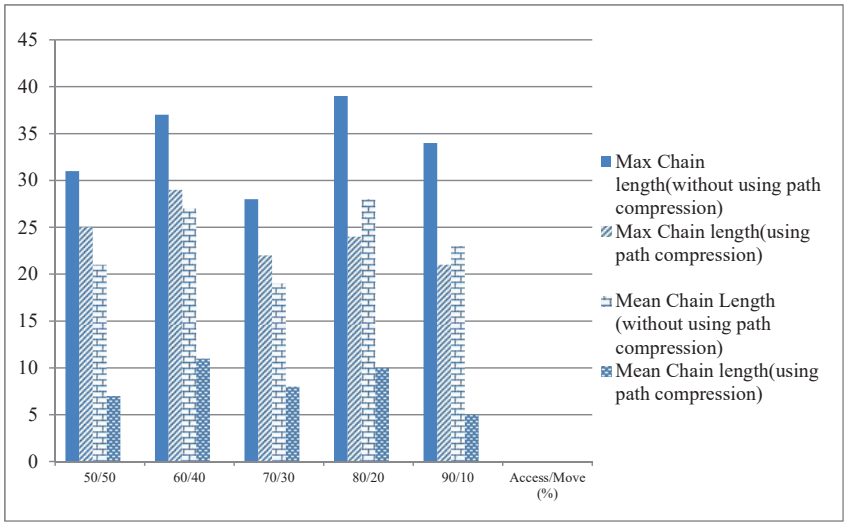


Figure 10 Maximum and mean chain length for T = 10 using binomial distribution.

with "Access," and the moving of the objects was shown with the "Move" operations.

In the distributed environment, system modeling was performed in two ways: static and dynamic. In the static environment, while access to the objects was performed frequently, the moving of the objects was performed less. In contrast, in the dynamic environment, while access to the objects was performed less, the moving of the objects was performed more frequently. In our study, modeling was carried out for some processes related to the objects in the static and the dynamic environments.

In Figures 10–12 the results of the distributed environment simulation according to the various Access/Move (%) rates, respectively, for T = 10, T = 20 and T = 30 are shown using the binomial distribution. In these graphics, the graph comparatively shows the maximum chain length and the average chain length for the cases in which we used and did not use the path compression algorithm. In this graph, **T** is the threshold value, and if the chain length is equal to or greater than the threshold value, the chain length is broken. In our study, by changing the **T** value, the appropriate values could be found. Each column in the graph shows the maximum and the average chain lengths in the cases in which we used and did not use the path compression algorithm, respectively. In the graph, the maximum and average chain lengths are calculated by increasing the amount of access to the objects via the

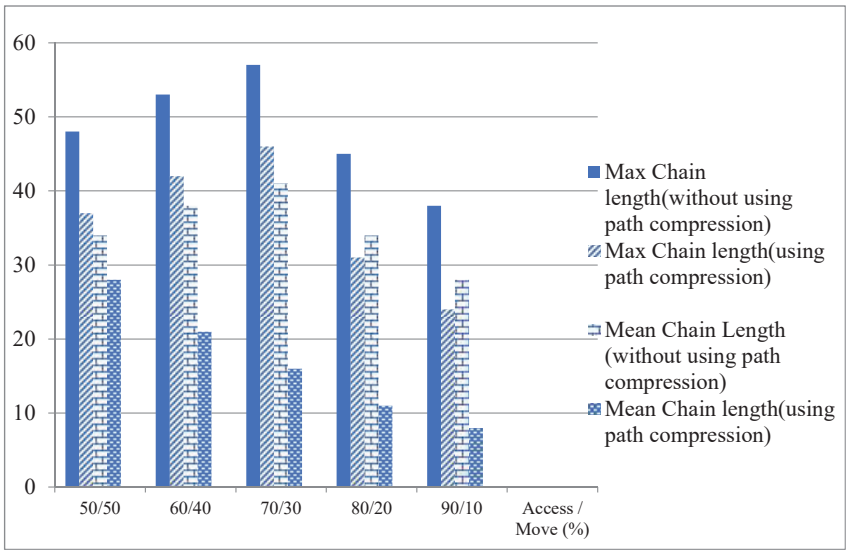


Figure 11 Maximum and mean chain length for T = 20 using binomial distribution.

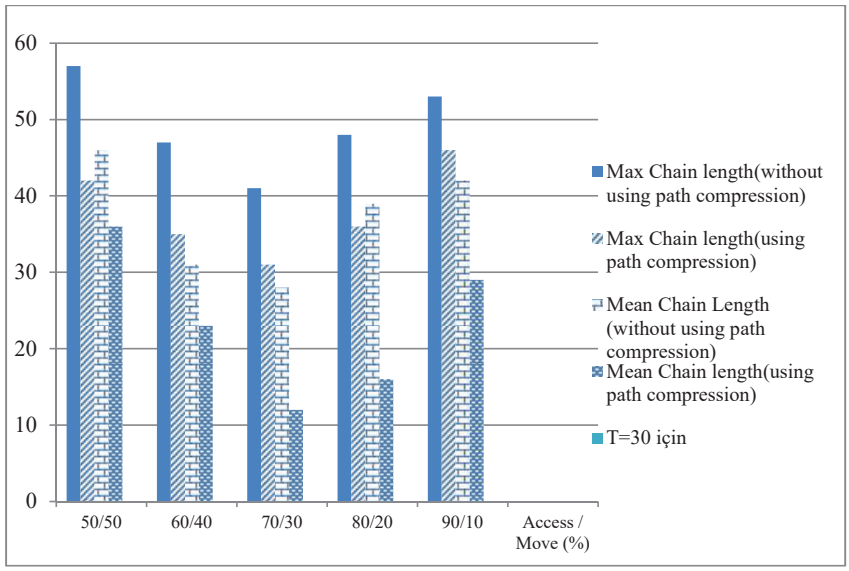


Figure 12 Maximum and mean chain length for T = 30 using binomial distribution.

Access/Move (%) rates, such as 50/50, 60/40, 70/30, 80/20, and 90/10, and by decreasing the number of moving objects. When we increased the amount of access to the objects and used the path compression algorithm, the average chain length was broken more.

6.2 Normal Distribution

In Figures 13–15 the results of the distributed environment simulation according to the various Access/Move (%) rates, respectively, for T = 10, T = 20 and T = 30 are shown using the normal distribution. In these graphics, the graph comparatively shows the maximum chain length and the average chain length for the cases in which we used and did not use the path compression algorithm. In this graph, **T** is the threshold value, and if the chain length is equal to or greater than the threshold value, the chain length is broken. In our study, by changing the **T** value, the appropriate values could be found. Each column in the graph shows the maximum and the average chain lengths in the cases in which we used and did not use the path compression algorithm, respectively. In the graph, the maximum and average chain lengths are calculated by increasing the amount of access to the objects via the Access/Move (%) rates, such as 50/50, 60/40, 70/30, 80/20, and 90/10, and by decreasing the number of moving objects. When we increased the amount

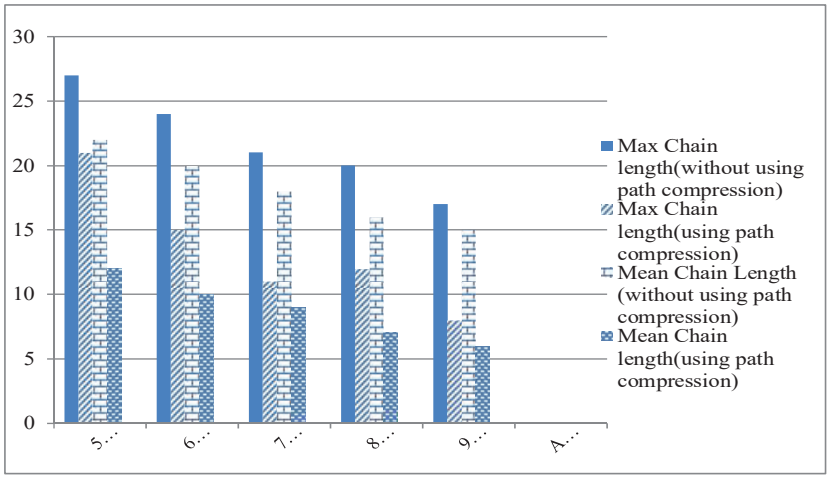


Figure 13 Maximum and mean chain length for T = 10 using normal distribution.

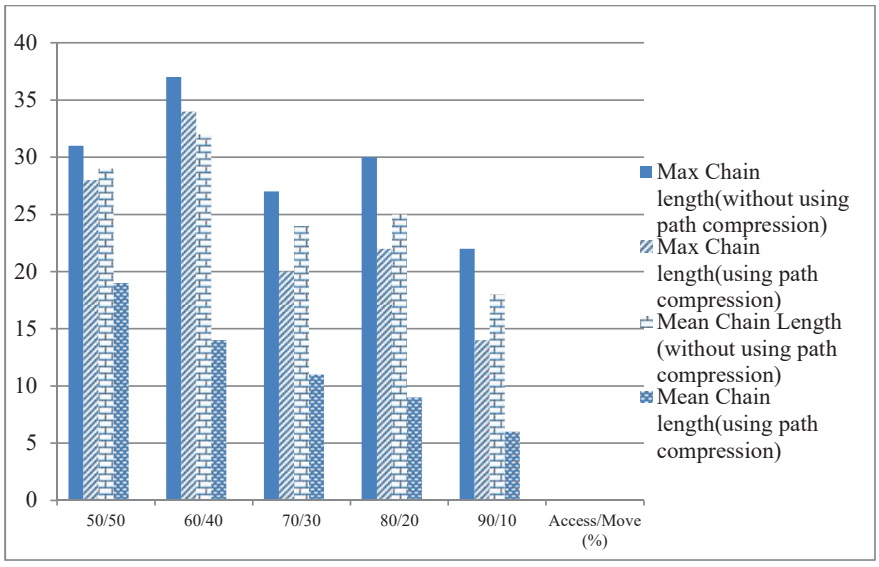


Figure 14 Maximum and mean chain length for T = 20 using normal distribution.

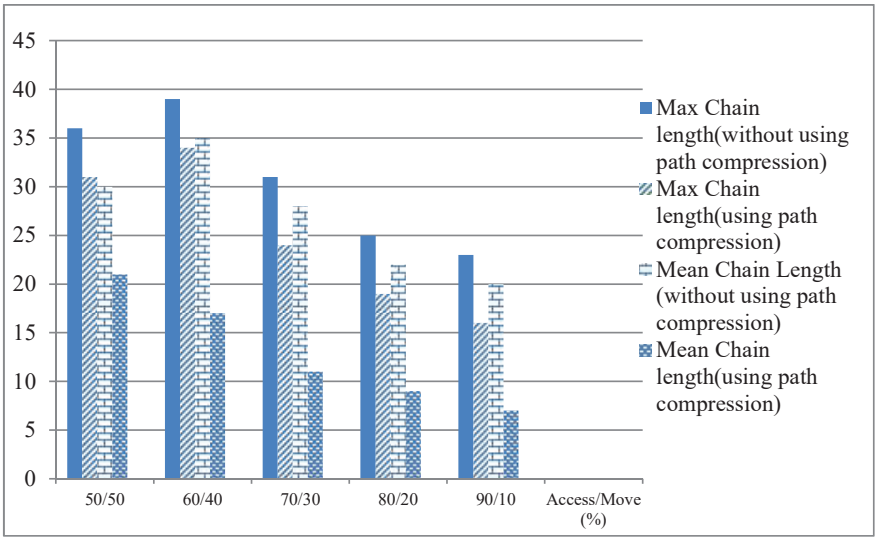


Figure 15 Maximum and mean chain length for T = 30 using normal distribution.

of access to the objects and used the path compression algorithm, the average chain length was broken more.

When we used the path compression algorithm, the maximum length of the chain decreased. In addition, when we used a normal distribution, the length of all the objects was broken because access to all the objects was carried out. In contrast, the average length of the chain decreased dramatically as compared to the maximum length of the chain. Thus, using the path compression algorithm, we decreased the cost of access to the objects by breaking the long chain of nodes created by the objects, and fast access to the objects was achieved.

Moreover, unlike binomial distribution, when we use normal distribution, access to all objects are performed. Because access to all objects is made, the chain in which each object resides is broken. The average chain length decreases dramatically compared to the maximum chain length. This result shows that the path compression algorithm reduces the average access cost to objects by breaking the long node chain created by objects, and quick access to the objects is ensured.

7 Conclusion

In this paper, the data security problem in distributed databases was analyzed. In particular, the decentralized label model relevant to the data flow control was introduced, and its application methods were presented via examples. Furthermore, the flow of data objects in a decentralized environment was modeled using graphs.

In this study, data security problem in distributed databases, especially distributed label model related to data flow control is used. There are three basic concepts as being actor, object and label for the implementation of security and privacy policies. Each actor sets its own security and privacy policy by labelling the data. Thus, the data confidentiality, privacy, data integrity and data consistency in data transmission are provided.

In our study, the operations for giving and receiving of the authority between actors are performed. In our study, there is no separate authority or access control for each process such as reading, writing, updating, deleting. Access control and authority operation are done through labels. Unlike previous studies, data security is provided for all operations performed in the distributed database. Actors can revoke authority granted by itself or can give an authority to actor it wish at any time. The difficulties which occurred

during the implementation of security policies to distributed databases will be overcome.

In previous studies, a separate label was used for each operation (read, write) performed on the object and only read and write operations were performed. In the study recommended by us, all operations performed on the object (read, write, update, delete) are performed using a single label. This shows that the model we recommend is flexible. It is attempted to prevent the information disclosure by following-up the access of malicious actors to the data.

In these days, the data security is an important problem in many institutions such as banks and companies. This study has brought an important approach to the solution of this problem. In the continuation of this study, a prototype application showing the work of the label model will be established and the model will be enriched with re-labelling which also takes into account the hierarchy of actors. In addition, it is aimed to expand the work by eliminating certain deficiencies in the actor hierarchy.

While accessing the objects, a long chain might be formed between nodes. Via the path compression method, this chain was broken, the cost of accessing objects was lowered, and the performance gain was attained. The simulation of the distributed environment was carried out through an experimental study, and in addition to the evaluation of the algorithm, its effectiveness was demonstrated by the experimental study.

In future studies, a prototype application displaying the working of the label model will be presented, and the model will be enriched via relabeling that takes into account the hierarchy among principals. As a future study, a prototype application showing the work of the label model will be created and the model will be enriched with the re-labeling that takes into account the hierarchy of actors.

References

[1] Lin, J.; Yu, W.; Zhang, N. A survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy and Applications, IEEE Internet of Things, 2017, 4, 5, pp. 1125–1142.

[2] Mercuri, R. T. The HIPAA-Potamus in Health Care Data Security, Communications of the ACM Security Watch, 2004, 47, 2, pp. 25–28.

[3] Dağdeviren, M.; Dönemez , N. Developing a new model for Supplier Evaluation Process for a company and its Applications, Journal of

Faculty of Engineering and Architecture of Gazi University, 2006, 21, 2, pp. 247–255.

[4] Sultana, T.; Almogren, A. Data Sharing System Integrating Access Control Mechanism using Blochchain-Based Smart Cantracts for IoT Devices, MDPI Applied Sciences, 2020, 10, 2, pp. 488–509.

[5] Vural, Y.; Sağıroğlu, Ş. A review on Enterprise Information Security and Standards, Journal of the Faculty of Engineering and Architecture of Gazi University, 2008, 23, 2, pp. 507–522.

[6] Amo, D.; Alien, M. Protected Users: A Moodle Plugin to Improve Confidentiality and Privacyy Support through User Aliases, MDPI Sustainability Opportunities and Challenges for the Future of Open Education, 2020, 12, 6, pp. 2548–2564.

[7] Bogaert, K. Confidentiality and Privacy: What is the difference?, South African Family Practice, 2009, l 51, 3, pp. 194–195.

[8] Vimercati, S.; Foresti, S.; Livraga, G. Privacy in Pervasive Systems: Social and Legal Aspects and Technical Solutions, Data Management in Pervasive Systems, 43–65, 2015.

[9] Papadimitriou, P.; .Garcia-Molina, H. Data Leakage Detection, IEEE Transactions on Knowledge and Data Engineering, 2020, 23, 1, pp. 51–63.

[10] Faria, P.L.; Cordeiro, J.V. Health data privacy and confidentility rights: Crisis or redemption?, Springer Revista Portuguesa de Saute Publica, 2014, 32, 2, pp. 123–133.

[11] Clifton, C.W. Privacy Beyond Confidentiality, In Procedings of the ACM SIGSAC Conference and Communications Security (CCS'14), pp. 1156–1156, 2014.

[12] Hurrah, N.; Parah, S. ecure data transmission framework for confidentiality in IoT", Elsevier Ad Hoc Networks, 2019, 95, 101989.

[13] Al-Jarabi, S.; Al-Shourbaji, M.S. Survey of main challenges (security and privacy) in wireless body area networks for healtcare applications, Egyptia Informatics Journal, 2017, 18, pp. 113–122.

[14] Esfandiari, H.; Hajigohayi, M. Streaming Algorithms for Estimating the Matching Size in Planar Graphs and Beyond, ACM Transactions on Algorithms, 2018, 14.

[15] Liu, J.; George, M. D. Fabric: A Platform for Secure Distributed Computation and Storage, ACM Symposium on Operating Systems Principles and Implementation (SOSP), pp. 321–334, 2009.

[16] Clifton, W. Privacy Beyond Confidentiality, In Procedings of the ACM SIGSAC Conference and Communications Security (CCS'14), 2014, pp. 1156–1156.
[17] Al-Jarabi, S.; Al-Shourbaji, M.S. Survey of main challenges (security and privacy) in wireless body area networks for healtcare applications, Egyptia Informatics Journal, 2017, 18, pp. 113–122.
[18] Gupta, B.B.; Shingo, Y. Advances in Security and Privacy of Multimedia Big Data in Mobile and Cloud Computing, Multimedia Tools and Applications, 2018, 77, 7, pp. 9203–9208.
[19] Vorakulpipat, C.; Sirapaisan, S. A Policy-Based Framework for Preserving Confidentiality in BYOD Environments. A review of Information Security Perspectives, Hindawi Security and Communication Networks, 2017, pp. 1–11.
[20] Liu, J.; Arden, O. Fabric:Building Open Distributed Systems Securely by Construction, Journal of Computer Security, 2017, 25, 4–5, pp. 367–426.
[21] Garesh, N.; Troia, F. Static Analysis of Malicious Java Applets, ACM on International Workshop on Security and Privacy Analytics, Lousiana USA, pp. 58–63, 2016.
[22] Kim, N.Y.; Ryu, J.H. CF- CloudOrch:Container fog node-based cloud orchestration for IoT networks, The Journal of Supercomputing, 2018, 74, 12, 7024–7045.
[23] Bakir, Ç.; Hakkoymaz, V. Dağıtık Veritabanında Veri Etiketleme ile Bilgi Akış Denetimi, 5.Ulusal Yüksek Başarımlı Hesaplama Konferansı, Esenler İstanbul, 1–6, 2017.
[24] Cai, F.; Zhu, N. Survey of Access Control Models and Technologies for Cloud Computing, Springer Cluster Computing, pp. 1–12, 2018.
[25] Atlam, H.; Alassafi, M.O. XACML for Building Access Control Policies in Internet of Things, 3rd International Conference on Internet of Things, Big Data and Security, pp. 253–260, 2018.
[26] Rana, M.; Jayabalan, M. Privacy and Security Challenges towards Cloud Based Access Control in Electronic Health Records, Asian Journal of Information Technology, 2017, 16, 2–5, pp. 274–281.
[27] Alrumayh, A.;Lehman, S.Context aware access control for home voice assistent in multi-occupant homes, Pervasive and mobile computing, 2020.
[28] Mistry, I.; Tanwar, S. Blockchain for 5G-enabled IoT for industrial automation: A Systematic review, solutions and challenges, Elsevier Mechanical Systems and Signal Processing, 2019, 135, 106382.

[29] Servos, D.; Osborn, S.L. Current Research and Open Problems in Attribute-Based Access Control, ACM Computing Surveys, 49, 4, 2017.

[30] Li, Q.; Snadhu, R. Mandatory Content Access Control for Privacy Protection in Information Centric Networks, IEEE Transactions on Dependable and Secure Computing, 2017, 14, 5, pp. 494–506.

[31] Viswasrao, D.;Kumar, A. Blochchain-enabled Distributed Security Framework for Next Generation IoT: An Edge- Cloud and Software Defined Network Integrated Approach, IEEE Internet of Things Journal, 2020, pp. 1–8.

[32] W. Cheng, D.R. Ports at all, "Abstractions for Usable Information Flow Control in Aeolus", 2012 Usenix Annual Technical Conference, pp. 1–13, 2012.

[33] R. Barejee, S. Chatterjee at all, "Performance of a Discrete Wavelet Transform Based Path Merging Compression Technique for Wireless Multimedia Sensor Networks", Wireless Personal Communications, vol. 4, pp. 57–71, 2019.

[34] J. Janet, S. Balahrishnan at all, "Optimizing Data Movement within Cloud Environment using Efficient Compression Techniques", International Conference on Information Communication and Embedded Systems, 2016.

[35] B. Liu, X. Yu at all, "Blockchain based Data Integrity Service Framework for IoT Data", IEEE International Conference on Web Services, 2017.

[36] J. Cui, L. Shao at all, "Data aggregation with end-to-end Confidentiliaty and Integrity for large-scale Wireless Sensor Networks", Peer to Peer Networking and Applications, 2018, 25, 5, pp. 1022–1037.

[37] Vijayakumar, K.; Arun, C. Continuous security assesment of cloud based applications using distributed hashing algorithm in SDLC", Springer Cluster Computing, 2019, 22, pp. 10789–108000.

[38] Myers, A. C.; Liskov, B. Complete, Safe Information Flow with Decentralized Labels, In Proc. IEEE Symposium on Security and Privacy, 1998.

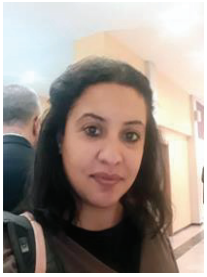
[39] Burow, N.; Carr, S.A. Control-Flow Integrity; Preccision, Security and Performance, ACM Computing Surveys, 50, 1, pp. 1–16, 2017.

[40] Ulaştırma Denizcilik ve Haberleşme Bakanlığı, Ulusal Siber Güvenlik Stratejisi, 2016–2019, Referans.

[41] Rong, X.; Hui, L. Provenance-based data flow control mechanism for Internet of things, Wiley Online Library Security and Privacy, 2020, pp. 1–23.

Biographies

Veli Hakkoymaz received B.S. degrees in computer engineering from Hacettepe University, in 1987 and M.S.degree in Computer Science from University of Pittsburgh (PA), In 1992, Ph.D.degree in CWRU (OH) in 1997. In 2011, he received the title of Associate Professor at Yildiz Technical University. He works Yildiz Technical University as Associate Professor since 2011. His research interests include database management systems, computer architecture, operating systems and distributed systems.



Cigdem Bakir is a Ph.D. student at the University of Yildiz Technical University since 2014. She received B.S. degrees in computer engineering from the University of Sakarya, in 2010 and the M.S. degree in computer engineering from Yildiz Technical University, İstanbul, in 2014. She worked a Research Assistant at Yildiz Technical University and Igdir University. She works Erzincan Binali Yildirim University since 2020. Bakir is currently completing a doctorate in Computer Science at the University of Yildiz Technical at Istanbul. Her Ph.D. research interests include information security, distributed database, and computer networks.