Hindawi
Scientific Programming
Volume 2021, Article ID 9913645, 8 pages
https://doi.org/10.1155/2021/9913645

Research Article

A Single-Label Model to Ensure Data Consistency in Information Security

Cigdem Bakir

Department of Computer Engineering, Yildiz Technical University, Istanbul 34220, Turkey

Correspondence should be addressed to Cigdem Bakir; cigdem.bakr@gmail.com

Received 5 March 2021; Revised 23 March 2021; Accepted 26 March 2021; Published 2 April 2021

Academic Editor: Yi-Zhang Jiang

Information security is defined as preventing actions such as unauthorized access and use, modification, and removal of information. It consists of certain basic elements of confidentiality, integrity, and accessibility. There are numerous studies in published literature which have been conducted to ensure information security. However, there is no previous study that covers these three basic elements together. In the present study, a model that includes these three key elements of information security together for big data was proposed and implemented. With this proposed "single-label model," a more practical and flexible structure was established for all operations (read, write, update, and delete) performed on a database on real data. In previous studies conducted with a label model, separate labels were used for read-only or write-only operations, and there was no structure that could ensure both confidentiality and integrity at the same time. The present study, however, shows what type of authorization and access control could be established between which processes and which users by looking at a single label for all the operations performed on the data. Thus, in contrast to the previous studies seen in published literature, data confidentiality, data integrity, and data consistency were all guaranteed for all transactions. The results of the proposed single-label model were also shown comparatively by conducting an experimental study of its application. The results obtained are promising for further studies.

1. Introduction

Information security is defined as preventing actions such as unauthorized access and use, modification, and removal of information, and it consists of certain basic elements including confidentiality, integrity, and accessibility [1, 2]. Confidentiality is the protection of information against being accessed, read, or used by unauthorized persons in any way. Integrity is the prevention of modification of information by unauthorized persons and the preservation of its original nature. Accessibility, on the other hand, is that the information is accessible and readily available as long as it is needed.

Today, there are new and highly effective threats that damage information systems and resources [3]. Although there are many measures taken to protect systems from such harmful threats that are supported by advanced technologies, it has been seen that attackers can still often succeed. In these and similar cases, any incident that causes a violation of any of the three basic elements of information security (confidentiality, integrity, and accessibility) is considered to be a security problem [4]. While some violations intentionally make systems inaccessible and disrupt services, others occur accidentally due to unforeseen faults. Whether accidental or malicious, security violations seriously affect the activity and reliability of an institution.

In general, threats often turn into attacks by exploiting gaps or vulnerabilities in systems. Therefore, it can be said that it is of great importance to provide all these three basic elements together to prevent such attacks from damaging information systems. In short, no matter how secure a system is, the important thing here is to ensure control of the access and authorization processes that may allow any attack [5].

Some leading factors that cause security breaches (or violations) include Denial of Service (DOS) attacks,

Distributed Denial of Service (DDOS) attacks, inappropriate web browsing behavior, wiretapping, access to resources using a backdoor, and data changes occurring accidentally or intentionally [6]. Data that is deliberately or accidentally changed directly affects the integrity principle of information systems security in particular, and it results in an emerging security breach. The occurrence of such data modification events, like giving excessive authorization to users and exercising poor control of permissions, plays an important role [7]. To deal with such problems, a model designed according to the specific access rights (e.g., read, write, update, and delete) is required for organizations and users. However, studies have shown that these models are unable to fully meet the needs of rapidly growing and increasingly complex systems, because they represent a serious financial burden and fail to fully provide information flow control [8–11]. Therefore, it is seen that it is not enough for information systems to be constructed in a way to protect them only from unauthorized access, malicious users, and misuse. In this study, a model was created to provide the three basic elements of information security together by using real data. In this way, no user or group of users would be able to access data that is not authorized at their level or data that they are not allowed to perform various operations on.

In this study, a single-label model is created. The scientific contribution of this model is that while the data available to be used by the stakeholders can only easily be used by authorized actors, it does not allow the use of these data by unauthorized third-party 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. Therefore, in this study, 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 this label model targets data confidentiality and integrity among nonreliable actors and environments. Through the labels given to the data, each actor can determine his/her own security policy independently from other actors and authorize the ones that he/she chooses. The purpose of this study was to develop a method that allows different users to access the data in a distributed environment and protects confidentiality. It was aimed at investigating methods preventing unauthorized access to data being accessed jointly by multiple actors.

In the remainder of this study, other researches related to this subject are presented in Section 2, while the method is presented in Section 3. The proposed model is discussed in Section 3, and its application is detailed in Section 4. Section 5 details the evaluation and conclusions.

2. Related Works

Information is a valuable asset. Therefore, access, processing, updating, deleting, and authorizing operations should be carefully managed to ensure that confidentiality, integrity, and accessibility are maintained. In recent years, some techniques have been developed in published literature which outline the rules related to access, authorization, monitoring, and control of information and information systems [12–14]. However, it is seen in many industries that the development area of these techniques has narrowed and that existing techniques do not fully meet the new business requirements that arise with developing technology, and they cannot be managed in accordance with the organizational structure. In addition, serious costs arise in the progress towards a manageable model, and the dynamism that is necessary for the use and sharing of resources is not achieved.

In recent years, various studies using different techniques for the purposes mentioned have been described in published literature. Schultz and colleagues developed a platform that allowed the data access of users to be automatically tracked. Because a user logs into the system separately for each transaction, authority control is performed again. The user has to perform the authority check at each stage. If he/she does not perform the check at any one stage, data confidentiality is breached. This creates the need for automatic monitoring of authority [15]. Parker et al. presented a platform extension for database transactions. In this platform, each table has a label and protects its length [16], but this method can impose high computational costs and high overheads. Yang et al. used information flow control in web applications, but this approach can be expensive in both space and time and requires more memory [17]. Muthukumaran et al. applied information flow control (IFC) with FlowWatcher monitoring software that provides applications with a web proxy but limits the granularity of policies it can enforce [18]. In previous studies in published literature [19–22], a separate label was used for each operation (read, write) carried out on the object, and only reading and writing were performed. In the present study's proposal, by contrast, all operations performed on the object (read, write, update, and delete) are carried out using a single label. In this way, by looking at a single label, what type of authorization style is used between which operations and which actors can be understood.

In recent years, various studies using different techniques for the purposes mentioned above have been described in published literature [13, 15, 23–26]. In this present study, on the other hand, there is no need for separate control for both authorizing and denying authorization. There is no need for separate authorization or access control for each operation such as reading, writing, updating, and deleting. In addition, by tracking the access of malicious actors to data, attempts are made to prevent information disclosure.

Fog computing or fog networking, also known as fogging, is pushing the frontiers of computing applications, data, and services away from a centralized cloud to a logical stream on the network edge. Fog networking systems work on building the control, configuration, and management over the Internet backbone [27].

Software-defined networking (SDN) is a promising approach to networking which provides an abstraction layer for the physical network [28]. In published literature, a recurrent neural network (RNN) model based on a new

regularization technique (RNN-SDR) was proposed by the authors. This technique supported intrusion detection within SDNs [28]. Nevertheless, this model is not practical for implementation in the context of an SDN. Prete and Schweitzer contextualized the existing problems in current computer networks and presented the SDN network as one of the main proposals for the viability of the Internet of the future. Simulations were created in an SDN network scenario using a POX Controller [29]. However, there is a need to obtain a synergistic effect that will make cloud environments more efficient, dynamic, and flexible, including automatic reconfiguration of network clusters.

In the current study, a single-label model was developed. The scientific contribution of this model is that while the data available to use by the stakeholders can be used easily only by authorized actors, it does not allow the use of these data by unauthorized third-party actors. At the same time, this model contributes to research into methods that enable the use of jointly used resources without causing information leakage. Therefore, in this study, a single-label model was developed which can maintain data confidentiality and integrity with information flow control. The difference between this study and other studies with a single-label model is that it targets data confidentiality and integrity of users. 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. Moreover, access control and authorization are ensured in accordance with the actor's 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. The difference between this study and other studies is that it provides data confidentiality, data integrity, and data consistency together.

3. Proposed Model

The single-label model consists of actors, objects, and labels.

3.1. Actor. The actors include data owners and users or groups of users who perform operations such as granting and receiving data authorization. Each actor labels his/her data for data confidentiality and integrity. The label consists of a list of security policies that are provided by the actors. Each actor labels his/her data for data privacy. That is, a label is determined which is paired with a data object. In addition, each actor has the right to safely change these security policies separately. Figure 1 shows a sample actor 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, $X \longrightarrow Y$ stands for granting authority by X to the principal Y. If $X \longrightarrow Y$ and $Y \longrightarrow Z$, then $X \longrightarrow Z$ is also true.

3.2. Label. A label is a collection of policies that are created for the protection of data. That is, a label is determined which is paired with a data object. In addition, each actor has the right to safely change these security policies separately. This model was developed for unreliable actors and environments. All actors change their own policy independently of each other. The object consists of data to which authorization is granted or received by actors. The label consists of the list of security policies issued by actors. Each actor labels his/her data for data confidentiality. In addition, each actor separately has the authority to safely change these security policies.

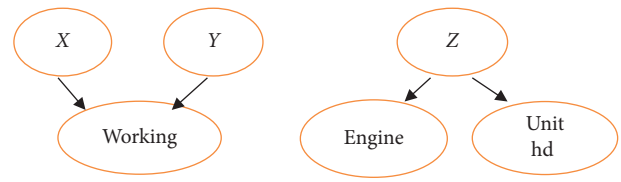

Figure 1: Examples of the principal hierarchy.

Figure 2 shows the contents of a label. Here, while u_1, u_2, ..., u_n show the owners of the data object from the actors in the system, the terms x_1, x_2, ..., x_m refer to the actors to whom authorization is given for any transaction by the data owners: $p_1, p_2, \ldots, p_n$, that is, each content definition on the L label, shows 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 objects to the other actors with its label.

3.3. Graph Modeling of Labels. In previous studies in published literature [19–22, 30–32], a separate label has been used for each operation (read, write) carried out on the object, and only reading and writing have been performed. However, in the present study proposal, all operations performed on the object (read, write, update, and delete) are carried out using a single label. In this way, by looking at a single label, what type of authorization style there is between which operations and which actors is understood.

In this present study, the single-label model is shown by a graph data structure (Figure 3) in which we let the label determined for graph G be L_G. In this study, the circles in the graph data structure show the actors. Which operation will be performed in the distributed database is determined by the way the arrow is drawn. A different arrow is used for each of the read, write, update, and delete operations. Thus, with a single label, a more practical and more secure authorization and access operation is created.

L_G consists of five parts, namely, owner, readers, writers, updaters, and deleters. The way the arrows are drawn in the graph show the types of authority needed to access the data. Here, while "owner" denotes the actors who own the labeled object, "readers" refers to the actors to whom authorization is given to read data owners' transactions; "writers" refers to the actors to whom authorization is given to write to the data owners' transactions; "updaters" refers to the actors to whom authorization is given to update the data owners' transactions; and "deleters" refers to the actors to whom

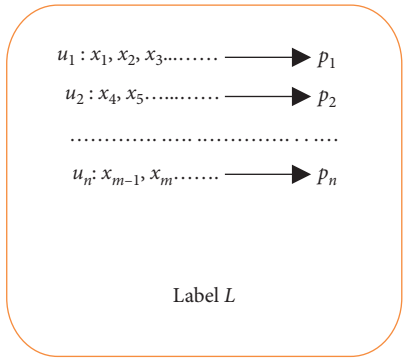

Figure 2: Label L example for the data object.

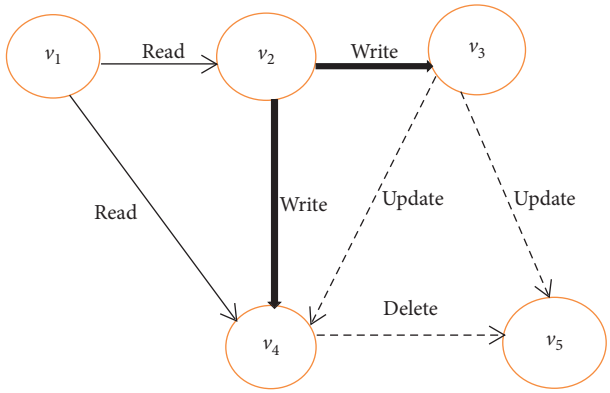

Figure 3: A graph G modeling of the label (L_G).

authorization is given to delete data owners' transactions. The label shown in Figure 1 combined with graph G can be expressed in the L_G typing format as follows:

$$L_G = \{v_1 : v_2, v_4; v_2 : v_3, v_4; v_3 : v_4, v_5; v_4 : v_5; v_5\}. \quad (1)$$

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 label belongs, v_2, v_3, v_4, and v_5 represent the actors authorized by the data owners for various object transactions (read, write, update, and delete).

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

The first policy is expressed with the $v_1 \longrightarrow v_1$, $v_1 \longrightarrow v_2$, and $v_1 \longrightarrow v_4$ 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 $v_2 \longrightarrow v_2$, $v_2 \longrightarrow v_3$, and $v_2 \longrightarrow v_4$ edges. This means that the v_2 actor allows the v_2, v_3, and v_4 actors to write to his/her data.

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

The third policy is expressed with the $v_3 \longrightarrow v_3$, $v_3 \longrightarrow v_4$, and $v_3 \longrightarrow v_5$ 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 $v_4 \longrightarrow v_4$ and $v_4 \longrightarrow v_5$ 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 $v_5 \longrightarrow v_5$ edge. This means that v_5 does not allow anyone other than himself/herself to perform any transaction on his/her data.

3.4. Bank Example. A bank has many customers. Each bank is obliged to protect and save its customers' account information such as money, goods, and investments from other customers or noncustomer principals. In Figure 4, a bank's customer operations have been shown by employing label modeling. In this figure, the oval shapes are as follows: M is customer, B is bank, and T is the principal's computing customer assets. Arrows represent information flow between principals, while squares represent the database and the data.

Any customer can, by labeling i ($1 \le i \le n$) assets with $\{M_i{:}B, M_i\}$, forge their own security policy. Also, each customer performs operations such as drawing or depositing cash and so forth at different times. A bank has to conduct these operations safely. These banks label all customer operations performed with $\{M{:}B, M\}$. Thus, banks can read customers' information. Customer i operations, like withdrawing cash, depositing cash, money transfers, and so forth, are conducted by the T principal. T is a program computing customers' asset details. The T principal can declassify any asset information that each i customer labels with $\{M_i{:}B, M_i\}$, and with a $\{B{:}B\}$ label it transfers them to the bank's database. Thus, this bank can control the flow of information and, to ensure that other principals in the system cannot read these data, it saves these data with a $\{B{:}B\}$ label in its private database. These labels are created for all operations performed in the database and combine them into one label.

4. Experimental Study

When the proposed single-label model was compared with the double-label model in published literature, and the performance results obtained in terms of accuracy and time are given in the following sections.

4.1. Accuracy. In Table 1, the success of the proposed single-label model and that of the double-label model in published literature are compared against a real data set, which has been taken from a hospital and whose classes are obvious. Accuracy rates were calculated for about 100 actors and 20 objects randomly selected from this data set. In addition, all classes of this data set were specified. Accuracy rates were calculated according to their real class. While measuring the accuracy rate, the classes of the model created for this study were calculated by comparing them with real classes. The success of the proposed model is clearly shown in Figure 5. When the performances of both methods were compared for all operations performed on objects in terms of accuracy

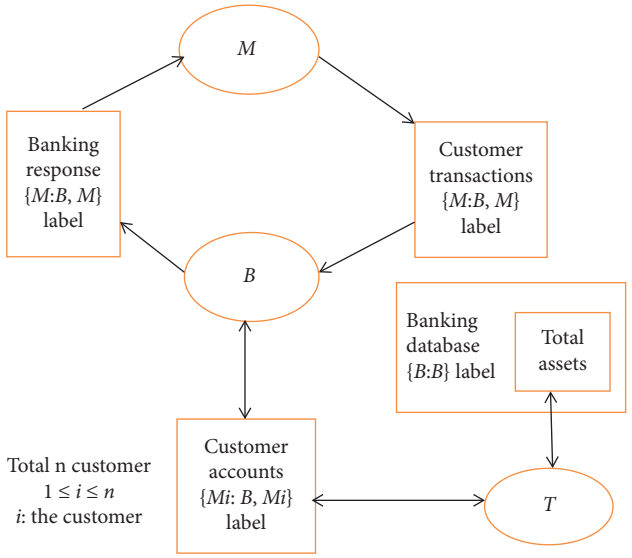

Figure 4: Labels for banking accounts.

Table 1: Accuracy rates for 100 actors and 20 objects.

Accuracy rate (%)	Double label	Single label
Read	87.27	99.94
Write	89.17	98.61
Update	79.50	96.37
Delete	83.34	97.81

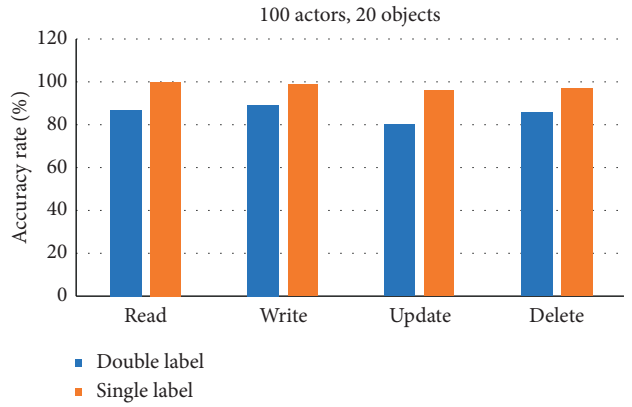

Figure 5: Accuracy rates for 100 actors and 20 objects.

rates, the success of the proposed single-label model can be clearly seen. In particular, it gives more successful results in reading and deleting operations. This is because writing and updating operations are more difficult than other operations.

In Table 2, the success of the proposed model (single label) and that of the model in published literature are compared in terms of accuracy. Accuracy rates have been calculated for about 1000 actors and 200 objects. The success of the proposed model is clearly shown in Figure 6. When the performances of both methods are compared in terms of accuracy rates for all operations performed on objects, the success of the proposed model can be clearly seen.

Table 2: Accuracy rates for 1000 actors and 200 objects.

Accuracy rate (%)	Double label	Single label
Read	84.21	96.93
Write	85.17	95.58
Update	82.96	91.58
Delete	81.55	95.10

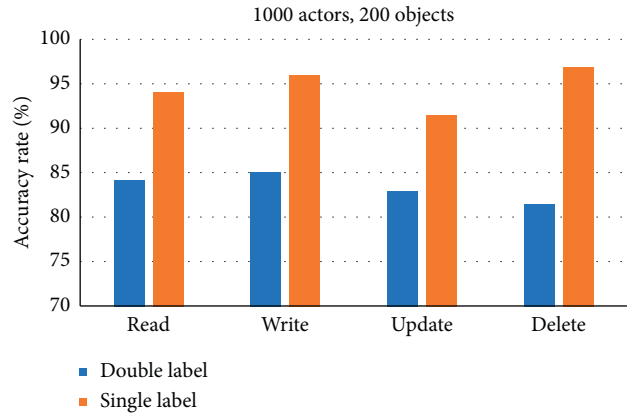

Figure 6: Accuracy rates for 1000 actors and 200 objects.

In Table 3, the success of the proposed model (single label) and that of the model in published literature are compared in terms of accuracy. Accuracy rates have been calculated for about 10000 actors and 2000 objects. The success of the proposed model is clearly shown in Figure 7. When the performances of both methods are compared in terms of accuracy rates for all operations performed on objects, the success of the proposed model can be clearly seen.

In Table 4, the success of the proposed model (single label) and that of the model in published literature are compared in terms of accuracy. Accuracy rates have been calculated for about 100000 actors and 20000 objects. The success of the proposed model is clearly shown in 8. When the performances of both methods are compared in terms of accuracy rates for all operations performed on objects, the success of the proposed model is clearly seen.

4.2. Time. In Table 5, the success of the proposed model (single label) and that of the model in published literature in terms of time are compared against the actual data set taken from the hospital. Performances related to time are given for about 100 actors and 20 objects. The success of the proposed model is clearly shown in Figure 9. In terms of time, it is seen that operations are performed on the data in less time with the proposed model. Writing and updating operations take longer in both methods in terms of time compared to other operations. This is because performing writing and reading operations on the object takes more time. Also, when compared in terms of time, the proposed model gives very successful results for all operations performed on the object.

In Table 6, the success of the proposed single-label model and that of the model in published literature in terms of time

Table 3: Accuracy rates for 10000 actors and 2000 objects.

Accuracy rate (%)	Double label	Single label
Read	75.63	92.64
Write	81.05	91.09
Update	74.87	89.75
Delete	83.78	90.56

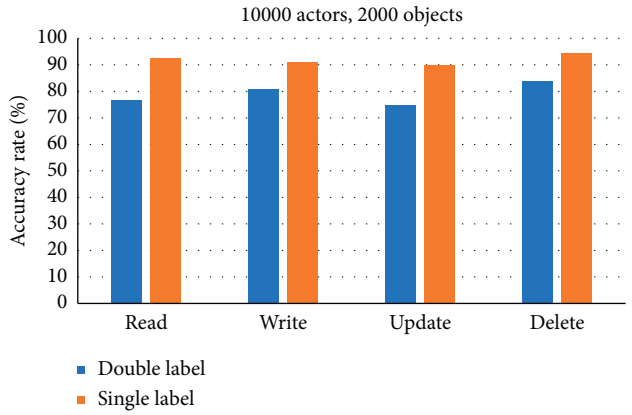

Figure 7: Accuracy rates for 10000 actors and 2000 objects.

Table 4: Accuracy rates for 100000 actors and 20000 objects.

Accuracy rate (%)	Double label	Single label
Read	70.50	88.64
Write	75.19	87.17
Update	72.65	81.60
Delete	81.64	86.38

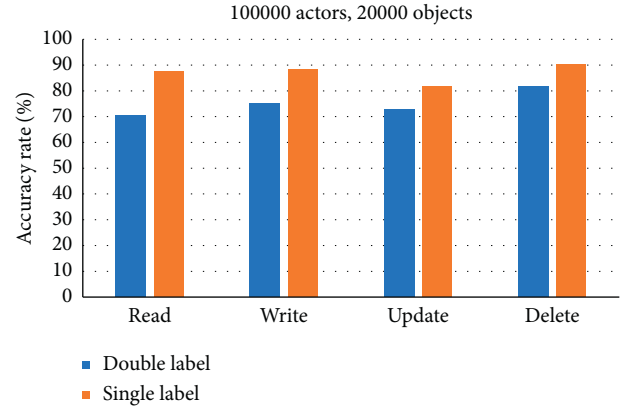

Figure 8: Accuracy rates for 10000 actors and 2000 objects.

are compared against the actual data set taken from the hospital. Performances related to the time are given for about 1000 actors and 200 objects. The success of the proposed model is clearly shown in Figure 10. In terms of time, it is seen that operations are performed on the data in less time with the proposed model.

In Table 7, the success of the proposed single-label model and that of the model in published literature in terms of time

Table 5: Times for 100 actors and 20 objects.

Time (sec)	Double label	Single label
Read	8.19	6.45
Write	9.82	7.97
Update	12.41	8.60
Delete	6.37	4.84

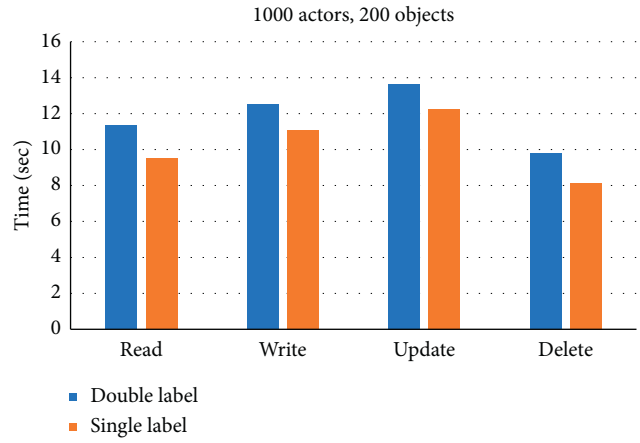

Figure 9: Times for 100 actors and 20 objects.

Table 6: Times for 1000 actors and 200 objects.

Time (sec)	Double label	Single label
Read	11.35	9.51
Write	12.51	11.09
Update	13.64	12.27
Delete	9.79	8.15

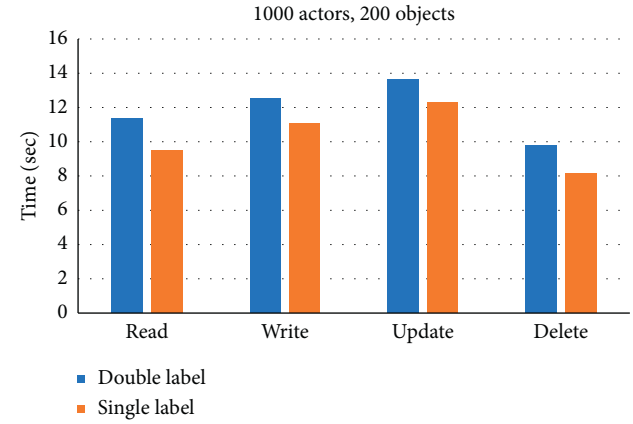

Figure 10: Times for 1000 actors and 200 objects.

are compared against the actual data set taken from the hospital. Performances related to the time are given for about 10000 actors and 2000 objects. The success of the proposed model is clearly shown in Figure 11. In terms of

Table 7: Times for 10000 actors and 2000 objects.

Time (sec)	Double label	Single label
Read	14.17	11.77
Write	15.02	12.82
Update	16.79	14.73
Delete	11.96	10.66

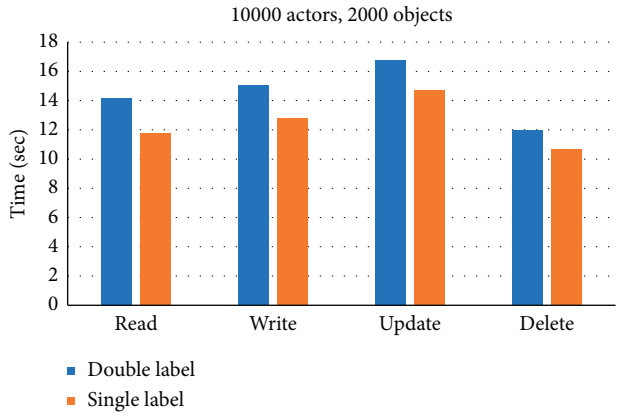

Figure 11: Times for 10000 actors and 2000 objects.

Table 8: Times for 100000 actors and 20000 objects.

Time (sec)	Double label	Single label
Read	15.72	14.35
Write	17.04	15.37
Update	17.78	16.14
Delete	12.01	11.02

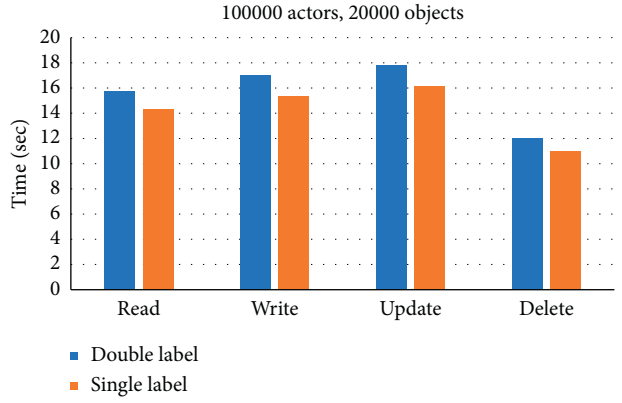

Figure 12: Times for 100000 actors and 20000 objects.

time, it is seen that operations are performed on the data in less time with the proposed model.

In Table 8, the success of the proposed single-label model and that of the model in published literature in terms of time are compared against the actual data set taken from the hospital. Performances related to the time are given for about 100000 actors and 20000 objects. The success of the proposed model is clearly shown in Figure 12. In terms of time, it is seen that operations are performed on the data in less time with the proposed model.

5. Evaluation and Conclusions

In this study, a single-label model was introduced for ensuring data security. In the proposed model, authorization and deauthorization operations between actors were both carried out. Also, in the proposed model, there is no separate authorization or access control for each operation such as reading, writing, updating, and deleting. Access control and authorization operations were performed through labels. Unlike previous studies, data security was ensured for all operations performed in the distributed database. Actors can take back the authority that they give at any time, or they can give authority to the actor they want. Challenges that occur during the implementation of security policies on distributed databases are overcome.

In this study, the problem of data security in distributed databases was addressed. In particular, a distributed-label model related to data flow control was introduced and examples of applications for its use were shown. In addition, data object flows in a distributed environment were modeled with a graph structure. In previous studies, a separate label has been used for each operation (read, write) carried out on the object, and only reading and writing have been performed. In the study proposed here, on the other hand, all operations performed on the object (read, write, update, and delete) were carried out using a single label. This also shows that the proposed model is flexible. By tracking the access of malicious actors to data, attempts were made to prevent disclosure of information. The results of the proposed single-label model for all operations performed on the data were also shown by the experimental study. It delivered more successful results, especially in reading and deleting operations.

The proposed model was also compared with the method used in previous studies in terms of time, and it was seen that it performed operations in a shorter time. In this way, data confidentiality, integrity, and consistency were ensured.

As a future study, a prototype application will be created, which shows the work of the label model, and the model will be enriched by relabeling, which takes into account the hierarchy of actors as well.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] A. Gupta and E. Galinkin, "Green lighting ML: confidentiality, integrity, and availability of machine learning systems in deployment"" in *Proceedings of the 37th International*

Conference on Machine Learning Workshop on Challenges in Deploying and monitoring Machine Learning Systems, pp. 1–10, Vienna, Austria, 2020.

[2] A. Tchernykh, U. Schwiegelsohn, E.-G. Talbi, and M. Babenko, "Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability," *Journal of Computational Science*, vol. 36, pp. 1–10, 2019.

[3] M. Jouini, L. B. A. Rabai, and A. B. Aissa, "Classification of security threats in information systems," *Procedia Computer Science*, vol. 32, pp. 489–496, 2014.

[4] M. Aminzade, "Confidentiality, integrity and availability-– finding a balanced IT framework," *Network Security*, vol. 2018, no. 2018, pp. 9–11, 2018.

[5] O. J. A. Pinno, A. R. A. Grégio, and L. C. E. De Bona, "ControlChain: a new stage on the IoT access control authorization," *Concurrency and Computation Practice and Experience*, vol. 32, pp. 1–23, 2019.

[6] B. A. Tama and K.-H. Rhee, "Data mining techniques in DoS/DDoS attack detection: a literature review," *Information*, vol. 18, no. 8, pp. 3739–3748, 2015.

[7] R. von Solms and J. van Niekerk, "From information security to cyber security," *Computers & Security*, vol. 38, pp. 97–102, 2013.

[8] N. Zeldovich, S. Boyd-Wickizer, and D. Mazieres, "Securing distributed systems with information flow control," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI '08)*, pp. 293–308, San Francisco, CA, USA, April 2008.

[9] M. Krohn, A. Yip, M. Brodsky et al., "Information flow control for standard OS abstractions," in *Proceedings of ACM Symposium on Operating Systems Principles (SOSP'07)*, vol. 41, pp. 321–334, Stevenson, WA, USA, October 2007.

[10] N. Zeldovich, S. Boyd-Wickizer, E. Kohler, and D. Mazières, "Making information flow explicit in HiStar," *Communications of the ACM*, vol. 54, no. 11, 2011.

[11] I. Roy, D. E. Porter, M. D. Bond, K. S. Mckinley, and E. Witchel, "Laminar: practical fine-grained decentralized information flow control," in *Proceedings of the 2009 ACM SIGPLAN Conference on Programming Language Design and Implementation in PLDI'09*, Dublin, Ireland, June 2009.

[12] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, and K. Sakurai, "Authentication in mobile cloud computing: a survey," *Journal of Network and Computer Applications*, vol. 61, pp. 59–80, 2016.

[13] M. Rana, M. Kubbo, and M. Jayabalan, "Privacy and security challenges towards cloud based access control in electronic health records," *Asian Journal of Information Technology*, vol. 16, no. 2–5, pp. 274–281, 2017.

[14] Q. Li, R. Sandhu, X. Zhang, and M. Xu, "Mandatory content access control for privacy protection in information centric networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 494–506, 2017.

[15] W. Cheng, R. K. Ports, and D. Schultz, "Abstractions for usable flow control in aelous," in *Proceedings of the USENIX Conference on Annual Technical Conference (USENIX ATC'12)*, pp. 1–12, Berkeley, CA, USA, June 2012.

[16] J. Parker, N. Vazou, and M. Hicks, "LWeb: information flow security for multi-tier web applications," *Proceedings of the ACM on Programming Languages*, vol. 3, pp. 1–30, 2019.

[17] J. Yang, T. Hance, and T. H. Austin, "Armando solar-lezama, cormac flanagan, and stephen chong, precise, dynamic information flow for database-backed applications," in *Proceedings of the 37th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 631–647, Santa Barbara, CA, USA, June 2016.

[18] D. Muthukumaran, O. 'K. Dan, C. Priebe, D. Eyers, B. Shand, and P. Peter, "FlowWatcher: defending against data disclosure vulnerabilities in web applications," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*, Seoul, Republic of Korea, October 2015.

[19] E. Cecchetti and A. C. Myers, "Nonmalleable information flow control," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*, pp. 1875–1891, Dallas, TX, USA, October 2017.

[20] J. Liu and O. Arden, "Fabric: building open distributed systems securely by construction," *Journal of Computer Security*, vol. 25, no. 4-5, pp. 367–426, 2017.

[21] A. C. Myers and B. Liskov, "Protecting privacy using the decentralized label model," *ACM Transactions on Software Engineering and Methodology*, vol. 9, no. 4, pp. 410–442, 2000.

[22] D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control," *ACM Computing Surveys*, vol. 49, no. 4, 2017.

[23] J. Liu and M. D. George, "Fabric: a platform for secure distributed computation and storage," in *Proceedings of the ACM Symposium on Operating Systems Principles and Implementation (SOSP)*, pp. 321–334, Koblenz, Germany, October 2009.

[24] N. Burow, S. A. Carr, J. Nash et al., "Control-flow integrity," *ACM Computing Surveys*, vol. 50, no. 1, pp. 1–33, 2017.

[25] Q. Aafaf, M. Hajar, A. E. Anas, and A. Q. Abdellah, "Access control in the internet of things: big challenges and new opportunities," *Elsevier Computer Networks*, vol. 12, pp. 237–262, 2017.

[26] J. B. D. Joshi, E. Latif, and A. Ghafoor, "A generalized temporal role-based access control model," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 1, pp. 4–23, 2005.

[27] M. R. Anawar, S. Wang, M. Azam Zia, A. K. Jadoon, U. Akram, and S. Raza, "Fog computing: an overview of big IoT data analytics," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 7157192, 22 pages, 2018.

[28] M. A. Albahar, "Recurrent neural network model based on a new regularization technique for real-time intrusion detection in SDN environments," *Security and Communication Networks*, vol. 2019, Article ID 8939041, 9 pages, 2019.

[29] L. Rodrigues Prete and C. M. Schweitzer, "Simulation in an SDN network scenario using the POX Controller," in *Proceedings of the 2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*, Bogota, Colombia, June 2014.

[30] A. C. Myers and B. Liskov, "Complete, safe information flow with decentralized labels," in *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, pp. 1–12, Oakland, CA, USA, May 1998.

[31] O. Arden and A. C. Myers, "A calculus for flow-limited authorization," in *Proceedings of the 2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pp. 135–149, Lisbon, Portugal, June 2016.

[32] A. C. Myers and B. Liskov, "A decentralized model for information flow control," *ACM SIGOPS Operating Systems Review*, vol. 31, no. 5, pp. 129–142, 1997.