Risk-Based Illegal Information Flow Detection in the IIoT

Argiro Anagnostopoulou[1][a], Ioannis Mavridis[2][b] and Dimitris Gritzalis[1][c]
[1]*Dept. of Informatics, Athens University of Economics and Business, Patision 76 Ave, Athens, Greece*
[2]*Dept. of Applied Informatics, University of Macedonia, 156 Egnatia St, Thessaloniki, Greece*

Keywords: Access Control, Information Flow Control, Illegal Information Flow Detection, Critical Infrastructure Protection, Industrial Internet of Things (IIoT), Industry 4.0.

Abstract: Industrial IoT (IIoT) consists of a great number of low-cost interconnected devices, including sensors, actuators, and PLCs. Such environments deal with vast amounts of data originating from a wide range of devices, applications, and services. These data should be adequately protected from unauthorized users and services. As IIoT environments are scalable and decentralized, the conventional security schemes have difficulties in protecting systems. Information flow control, along with delegation of accurate access control rules is crucial. In this work, we propose an approach to assess the existing information flows and detect the illegal ones in IIoT environments, which utilizes a risk-based method for critical infrastructure dependency modeling. We define formulas to indicate the nodes with a high-risk level. We create a graph based on business processes, operations, and current access control rules of an infrastructure. In the graph, the edges represent the information flows. For each information flow we calculate the risk level. This aids to reconstruct current access control rules on the high-risk nodes of the infrastructure.

1 INTRODUCTION

Internet of Things (IoT) describes ubiquitous connectivity to the Internet, turning common objects into connected devices, also called smart objects. Smart objects can sense the surrounding environment, transmit and process acquired data, and then provide feedback to the environment (Sisinni, 2018). However, once a device is connected to the Internet, it is vulnerable to cyber-attacks. The fourth industrial revolution (Industry 4.0), the exponential increase in connected devices worldwide, and the rapidly increasing number of cyber security incidents highlight the need for enhancing cyber resilience (ENISA, 2018).

In large scale Industrial IoT (IIoT) environments, all these interconnected devices must be protected against unauthorized access. Thus, there is need for the enforcement of proper access control policies on IIoT objects. This is the role of access control systems. Access control systems establish which active entities (subjects) are authorized to gain access to passive entities (objects), e.g., a resource. Each subject is characterized by a set of permissions for a specific object (Samarati, 2001), (Nakamura, 2019), (Jaune, 2011). Only authorized subjects are allowed to manipulate objects in authorized operations (Nakamura, 2019). The interconnected devices exchange data, creating information flows. However, access control mechanisms cannot always control how the information is used once it has been accessed. For this reason, information flow control (detection) is rising. Information flow control ensures that data contained in an object cannot flow into another and thus, these data cannot be accessed by users that do not have the appropriate permission (Jaume, 2011).

1.1 Contribution

In this work, we intend to address the improper or insufficient selection of access control rules that results to illegal information flows. Working on this direction, we propose an approach to detect and assess illegal information flows, which utilizes a risk-based

[a] https://orcid.org/0000-0003-4199-6257
[b] https://orcid.org/0000-0001-8724-6801
[c] https://orcid.org/0000-0002-7793-6128

Anagnostopoulou, A., Mavridis, I. and Gritzalis, D.
Risk-Based Illegal Information Flow Detection in the IIoT.
DOI: 10.5220/0012079800003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 377-384
ISBN: 978-989-758-666-8; ISSN: 2184-7711

analysis method for critical infrastructure dependency modeling (Kotzanikolaou, 2013a), (Kotzanikolaou, 2013b). We focus our proposed work in IIoT environments. Our approach creates a graph based on business processes, operations, and current access control rules of an infrastructure. In the graph, the edges represent the information flows. Our aim is to detect the high-risk nodes and thus, the nodes that are susceptible to participate in illegal information flows. Specifically, we compare the transactions that take place with the current access control rules, in order to mark the transaction that the information illegally flows from one node to another. Finally, we calculate the risk to each information flow. The calculated risks indicate the nodes that need reconstruction of their current access control rules.

1.2 Structure

The remainder of the paper is structured as follows. In Section 2, we present the related work regarding methodologies used for the detection of illegal information flows. In Section 3, we provide the theoretical background, including the key concepts that are used in the proposed approach. In Section 4, we present our approach for the illegal information flow detection in IIOT environments, while Section 5 provides a case study in which we evaluate the formulas for the risk calculation. Finally, the conlusion, limitations and future work are exhibited in Section 6.

2 RELATED WORK

Researchers and security professionals distinguish the terms of access control and information flow control. Access control verifies that subjects with specific access rights can manipulate an object, while information flow control tracks how information propagates through the program during execution (Hedin, 2012). In recent years, the detection of illegal information flows is a major topic of interest in scientific research. This section summarizes the main existing methods that focus on this topic.

Zimmermann et. al propose a policy-based intrusion detection approach to verify the legality of information flows created by system operations among objects. The flows that are not authorized by any security policy are considered as intrusion symptoms (Zimmermann, 2003).

Masri et al. present a dynamic information flow analysis to detect and debug insecure flows in programs. Authors incorporate a static pre-processing phase that detects both implicit flows at runtime, and the explicit ones. They also utilize a dynamic slicing algorithm that allows their approach to be applied to structured and unstructured programs (Masri, 2004).

Cai et. al investigate illegal information flows resulting from both roles and users. They obtain illegal information flows via the operations of difference, intersection, complement, etc. Authors propose, also, a strategy that focuses on the "writing" operation, as it is the primary means of creating information flows (Cai, 2009).

Hammer et. al propose a method based on program dependence graphs (PDG) to represent information flows in a program. The authors developed a dependence graph generator for full Java bytecode that requires less annotations than the traditional ones and provides more precise output. They, also, introduce flow equations, including the case of the declassification handling (Hammer, 2009).

Finally, Jaume et. al propose a framework that identifies illegal information flows. They present two implementations: Blare and JBlare. The first observes system calls and identifies information flows between OS containers, such as files or sockets. JBlare, an improved version of Blare, identifies information flows at the language level (Java) (Jaume, 2011).

Our approach assesses the information flows and detects the illegal ones in IIoT environments by utilizing a risk-based method for critical infrastructure dependency modeling (Kotzanikolaou, 2013a), (Kotzanikolaou, 2013b). Specifically, we model the business processes as a graph and assess the risk of all information flows to detect the nodes that are susceptible to participate in an illegal information flow. Currently, there is no research that utilizes a risk-based method for the detection of illegal information flows.

3 THEORETICAL BACKGROUND

This section provides a theoretical background, including the key concepts of information flow control and access control.

3.1 Access Control

Access control manages the acceptable operations for a user or process on system resources. The main concepts of an access control system are policies, models, and mechanisms. Access control policies specify how and when a user, or a process, may access a resource. An access control system enforces the access control policies through access control mechanisms, which are responsible for granting or denying access (Salonikias et. al, 2019).

There are several access control models/schemes that an organisation can adopt to enforce policies and allocate access rights to its subjects. Below we exhibit some of the most widely used ones.

Lattice-Based Access Control (LBAC): This scheme is the first to be developed to secure information flows in information systems. In a system, information flows from one object to another. An object can be defined as a container of information, such as files and directories in an operating system, or relations in a database. Information flow is typically controlled by assigning a security class to every object. Whenever information flows from object a to object b, there is also information flow from the security class of a to the security class of b (Sandhu, 1993), (Denning, 1976).

Mandatory Access Control (MAC): This scheme is based on lattice model of security level. It is based on two prominent rules: (i) No-read-up, and (ii) No-write-down. The first describes that a low-level subject is not allowed to read high-level objects. The latter describes that high-level objects can only be written by low-level subjects. As a result, in MAC scheme the information flows from lower levels to the higher ones (Bell et. al, 1972).

Discretionary Access Control (DAC): The principle of this scheme is that access to objects can be restricted based on the identity of the subjects and/or groups to which they belong. Subjects are the entities which cause an information flow between objects. Each object has a subject as an owner. The access policy of an object is determined by its owner. An owner can transfer information access to other subjects (Downs et. al, 1985).

Role-Based Access Control (RBAC): In this scheme the permissions are linked to roles. Users are assigned to appropriate roles based on their responsibilities and qualifications. Roles are created for job functions in an organisation. Users can be reassigned from one role to another. Roles can be granted new permissions as new applications and systems are incorporated, while permissions can be revoked from roles when is necessary (Sandhu et. al, 1996), (Salonikias et. al, 2019).

Capability-Based Access Control (CapBAC): In this model an owner of a device issues a capability token. This token is a set of access rights to a subject. The subject is allowed to manipulate the device based on the access rights that are defined in the capability token (Nakamura et. al, 2019).

3.2 Information Flow Control

An information system is composed of subjects and objects. An object contains data or operations to manipulate the data, such as databases or files. A subject is an entity, i.e., user or transaction, that manipulates an object. In order a subject to access data, it issues an operation to the respective object. A transaction refers to a sequence of operations on an object (Nakamura, 2018).

It is crucial for information security to define proper access control rules, and tracks how information is propagated by computing systems during execution. Thus, information flow control aims to enhance both the confidentiality and the integrity of the information (Hedin et. al, 2012).

Let assume that an object o supports one of the basic operations OP (e.g., read or write). An access rule is composed of a tuple <s, o, op>, while the pair <o, op> is called access right. An authorizer grants an access right to a subject s. Subject s is allowed to manipulate the object o in an operation op only if s is granted an access right <o, op> (Nakamura et. al, 2018).

Now, let suppose that a subject s_i is granted with the access right <f,read> on an object f, and an access right <g,write> on object g. Suppose another subject s_j is granted with an access right <g,read>. Let assume that s_i reads data d in the object f and then writes data d to the object g. The s_j is not allowed to read data in the object f. However, the s_j can obtain data d in the object f by reading data d stored in the object g. As a result, information in the object f illegally flows into the s_j via the s_i and the object g (Nakamura et. al, 2018).

In this work, we use four distinct terms to explain illegal transactions from one object to another: (i) illegal read, (ii) illegal write, (iii) suspicious read, and (iv) impossible write. *Illegal read* occurs if and only if (iff) a transaction reads data that are contained in an object, in which the transaction does not have the permission to access. *Suspicious read* occurs iff the transaction reads data in the object whose data is not allowed to be brought to other objects. *Illegal write* occurs iff the transaction writes data to the object after illegally reading data in another object. Finally, *impossible write* occurs iff the transaction writes data to the object after suspiciously reading data in another object (Nakamura et, al, 2018). Figures 1 and 2 depict two examples of illegal transactions (Nakamura et. al, 2018).

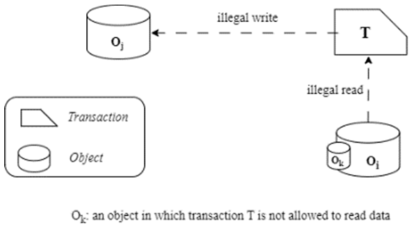
illegal write
T
O_j
illegal read
Transaction
Object
O_k
O_i
O_k: an object in which transaction T is not allowed to read data

Figure 1: Illegal read and Illegal write transactions (Nakamura, 2018).

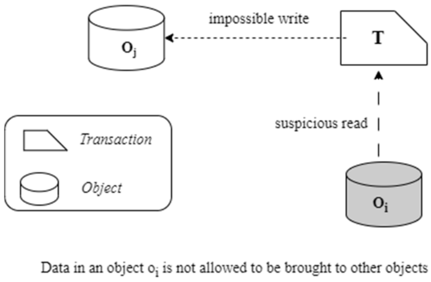
impossible write
T
O_j
suspicious read
Transaction
Object
O_i
Data in an object o_i is not allowed to be brought to other objects

Figure 2: Suspicious read and impossible write transactions (Nakamura et. al, 2018).

4 THE PROPOSED ILLEGAL INFORMATION FLOW DETECTION APPROACH

The proposed approach utilizes a risk-based method for critical infrastructure dependency modeling (Kotzanikolaou, 2013a), (Kotzanikolaou, 2013b) to assess information flows and detect the illegal ones in IIoT environments. We aim to find out the high-risk nodes that are prone to take part in illegal information flows.

The resulting method includes five steps:

Step 1. Dependency Graph Definition. The graph depicts information flows and thus the transactions between the nodes of the IIoT network.

Step 2. Supporting Metrics Calculation. The supporting metrics are: (1) severity, (2) operation factor, and (3) legality.

Step 3. Illegal Information Flow Likelihood Calculation for graph connections.

Step 4. Transaction Impact Calculation.

Step 5. Illegal Information Flow Risk Calculation.

4.1 Dependency Graph Definition

In this step, we denote as:

- O: set of objects in the IIoT network,
- IF: set of information flows between objects, and
- T: the set of transactions among object nodes for each information flow.

Dependencies are modelled in directed, weighted graphs G = (V, E), where the nodes V represent components of an IIoT network and edges E represent information flows between them. The graph is directional to represent an information flow from one component to another within the IIoT network. An edge $O_x \rightarrow O_y$ depicts an information flow $IF_{x\rightarrow y}$ from Object O_x to Object O_y. Each information flow may be related to many connecting transactions $T_{(x\rightarrow y)_i}$. Each transaction depicts a "get" or a "write" operation to an object.

Based on bibliography (Nakamura, 2019), when a subject S_i performs a "get" operation to object O_x, i.e., $<O_x$, get$>$, and the information flows from an object O_x to an object O_y, the transaction is depicted as follows:

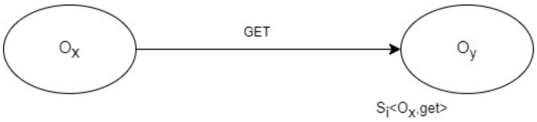

Figure 3: Depiction of "get" operation.

Similarly, we define that when a subject S_i performs a "write" operation to object O_x, i.e., $<O_x$, write$>$, and the information flows from an object O_x to an object O_y, this transaction is depicted as follows:

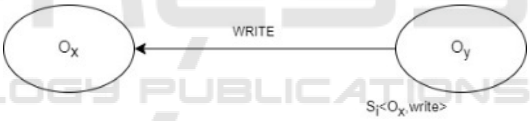

Figure 4: Depiction of "write" operation.

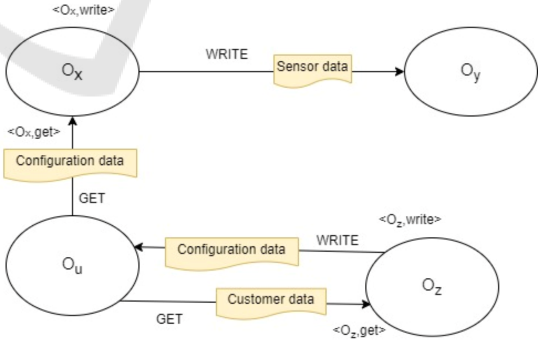

Figure 5: Example of a network dependency graph.

An indicative example of a network dependency graph is presented on Figure 5. The graph comprises four (4) objects: O_x, O_y, O_u and O_z. O_x writes Sensor data on O_y, while it gets Configuration data from O_u. Finally, O_z writes Configuration data to O_u, while it gets Customer data from O_u.

4.2 Supporting Metrics Calculation

This section introduces the metrics used in our approach to calculate the risk of each node so as to identify the nodes with the higher risk.

4.2.1 Severity

Depending on the data category that flows from one node to another, there is a corresponding *Severity*. This metric refers to both legal and illegal information flows. The range of Severity is [1,5], where 1 = very low, and 5 = very high.

We define six categories of data that usually exist in smart manufacturing. The catalogue with these categories may be enlarged or altered, according to the infrastructure that the method is applied. Table 1 presents the selected data categories and their values.

Table 1: Severity values based on data category.

Category of Data	Severity
Configuration data	5
Sensor data	4
Billing & Pricing data	3
Customer data	2
Meteorological data	1
Acknowledgement data	1

4.2.2 Operation Factor

Depending on the operation type (get, write) of the transaction, we define two distinct operation values for this metric. The value depends on the impact that the operation has to a specific transaction. We consider that the "write" operation may cause greater impact when the intent of the subject is malevolent. The term transaction defines the connection between two nodes and thus, the information flow between them. Table 2 presents operations and their values.

Table 2: Operation factor values based on operation type.

Type of Operation	Operation factor
get	2
write	3

4.2.3 Legality

The Legality metric characterizes a transaction based on six categories: (i) Legal Read, (ii) Legal Write, (iii) Illegal Read, (iv) Illegal Write, (v) Suspicious Read, and (vi) Impossible Write. The categories have been defined in section 3.2. The range of Legality is [1,5], where 1 = totally legal, and 5 = totally illegal. Table 3 depicts the legality values according to the category of the transaction that takes place.

Table 3: Legality values based on the transaction type.

Category of Transaction	Legality
Legal Read	1
Legal Write	1
Illegal Read	2
Illegal Write	3
Suspicious Read	4
Impossible Write	5

4.3 Illegal Information Flow Likelihood

Each relationship is assigned with a likelihood value, which declares how likely an illegal information flow is to occur. Intuitively, this value is a probability, based on which we can make predictions about the information flow state, at different times.

For each information flow $IF_{x \to y}$ from O_x to O_y, every transaction $T_{(x \to y)_i}$ is marked either as "Good" or as "Bad". A transaction is marked as "Good" when it is characterized as legal read or legal write. When it is characterized as illegal read, suspicious read, illegal write or impossible write, we mark it as "Bad". Formula 1 calculates the Illegal Information Flow Likelihood, denoted as $Likelihood_{x \to y}$. The range of $Likelihood_{x \to y}$ is [0, 1]. The parameter n is the total number of transactions between two nodes.

$$Likelihood_{x \to y} = \frac{\sum_{i=1}^{n}(T_{(x \to y)_i} \text{marked as "Bad"})}{\sum_{i=1}^{n}(T_{(x \to y)_i})} \quad (1)$$

4.4 Transaction Impact

Each transaction $T_{(x \to y)_i}$ is assigned with an impact value, denoted as $Impact_{(x \to y)_i}$. Since there is no standard available for evaluating the legality of information flows, we propose the following formula for the calculation of the $Impact_{(x \to y)_i}$.

$$Impact_{(x \to y)_i} = Severity * Operation\ Factor * Legality \quad (2)$$

When the value of $Impact_{(x \to y)_i}$ is high, a security incident may cause a serious impact to the operation of the IIoT environment.

We have thoroughly estimated all the possible combinations of the values for each parameter of the formula, and we concluded that the $Impact_{(x \to y)_i}$ *values* range between [2,75]. We rescale the range of the calculated $Impact_{(x \to y)_i}$ *values* into the range [1,10].

Rescaling helps all values to be in the same range. It adjusts the numbers to make it easy to compare the values of different metrics with different value ranges. Data rescaling assists in extracting inferences about the applicability and performance of an approach. Formula 3 depicts the function used for the rescaling of the *Impact*$_{(x\rightarrow y)_i}$ *values*:

$$f(I) = \frac{(b-a)(I-min)}{max-min} + a \tag{3}$$

where I is a particular *Impact*$_{(x\rightarrow y)_i}$ *value.*

Specifically, let assume that we want to scale a range [min, max] to [a, b]. We need a continuous function that satisfies the conditions: (i) f(min) = a, and (ii) f(max) = b. In our case min = 2, max = 75, a=0, and b=10. Table 4 assigns the ranges of the scaled *Impact* values to five concrete levels.

Table 4: Rescaling of the Impact values.

Calculated Impact Values	Scaled Impact Values	Scaled Impact Level
[2, 10)	[1,2)	Very Low
[10,26)	[2,4)	Low
[26,42)	[4,6)	Medium
[42,58)	[6,8)	High
[58,75]	[8,10]	Very High

Having calculating impact for each individual transaction $T_{(x\rightarrow y)_i}$, we evaluate the *Total Impact* value of an information flow $IF_{x\rightarrow y}$ from O_x to O_y as the average impact of the transactions $T_{(x\rightarrow y)_i}$ performed. Formula 4 calculates the *Total Impact* value of an information flow, where n denotes the number of transactions $T_{(x\rightarrow y)_i}$ between node x and node y.

$$Total\ Impact_{x\rightarrow y} = \frac{\sum_{i=1}^{n} Impact_{(x\rightarrow y)_i}}{n} \tag{4}$$

4.5 Illegal Information Flow Risk

The proliferation of impact and likelihood values indicate the illegal information flow Risk, denoted as $R_{x\rightarrow y}$, for an information flow $IF_{x\rightarrow y}$ from an object O_x to an object O_y as follows:

$$R_{x->y} = Total\ Impact_{x\rightarrow y} * Likelihood_{x\rightarrow y} \tag{5}$$

This metric identifies the information flows with the higher risk. Estimating risk helps identifying the flows and thus the nodes that are susceptible to illegal information flows. As a result, the nodes that participate in these flows need more accurate and stricter access control rules.

We establish a scale to determine whether the risk of a node is acceptable. The level of acceptability depends on the infrastructure and the criticality of its systems. In general, the lower the risk level, the more acceptable it is. Table 5 assigns the ranges of $R_{x\rightarrow y}$ values to five distinct levels.

Table 5: Levels of Illegal Information Flow Risk.

Illegal Information Flow Risk Value	Illegal Information Flow Risk Level
[0,2)	Very Low
[2,4)	Low
[4,6)	Medium
[6,8)	High
[8,10]	Very High

5 CASE STUDY

In this section we apply our method on an IIoT infrastructure network. Figure 6 presents a dependency graph which depicts how the network components (e.g., O_a, O_b, etc.) are connected, along with the category of data (e.g., configuration data, sensor data, etc.) that flow from an object to another. Each transaction refers its operation type (e.g., get, write). The illegal transactions are notated with the flag “Bad”, along with their category (e.g. illegal write, illegal read, suspicious read, impossible write).

Let assume that our interest focuses on object A. This object takes part in totally nine transactions. However, seven of them are marked as “Bad”. Thus, we suppose that O_a has a primarily malicious behaviour. Firstly, we calculate the *Illegal Information Flow Likelihood* for all the transactions of object A. The results are presented on Table 6.

Table 6: Illegal Information Flow Likelihood Calculation.

From	To	Illegal Information Flow Likelihood
O_a	O_b	*0.5*
O_a	O_c	*1*
O_a	O_f	*0*
O_a	O_g	*1*
O_a	O_h	*1*

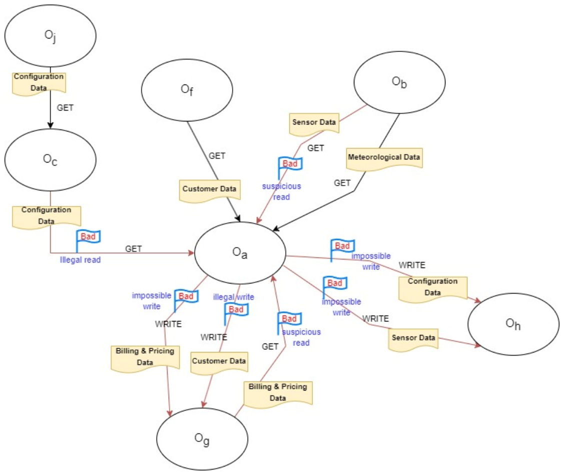

Figure 6: IIoT Infrastructure Network Dependency Graph.

For all the transactions we calculate the metrics of *Legality*, *Operation factor* and *Severity*. The values of these metrics are presented in Table 7.

Table 7: Calculation of the metrics Legality, Operation factor, and Severity.

From	To	Legality	Operation factor	Severity
O_a	O_b	*4*	*2*	*1*
O_a	O_b	*1*	*2*	*5*
O_a	O_c	*2*	*3*	*4*
O_a	O_h	*5*	*3*	*5*
O_a	O_h	*5*	*3*	*2*
O_a	O_g	*3*	*3*	*3*
O_a	O_g	*5*	*2*	*3*
O_a	O_g	*4*	*2*	*2*
O_a	O_f	*1*	*2*	*1*

Next step is the *Impact* calculation for all the transactions. Table 8 presents the values of this metric.

Table 8: Calculation of the Impact of all transactions.

From	To	Impact	Scaled Impact
O_a	O_b	*32*	*4.7*
O_a	O_b	*2*	*1*
O_a	O_c	*20*	*3.22*
O_a	O_h	*60*	*8.15*
O_a	O_h	*75*	*10*
O_a	O_g	*18*	*2.97*
O_a	O_g	*45*	*6.3*
O_a	O_g	*24*	*3.71*
O_a	O_f	*4*	*1.25*

Having calculating impact for each individual transaction, we evaluate the *Total Impact* value of an information flow $IF_{x\rightarrow y}$ from object O_x to object O_y as the average impact of the transactions $T_{(x\rightarrow y)_i}$ performed. The results are presented on Table 9.

Table 9: Calculation of Illegal Information Flow Impact.

From	To	Illegal Information Flow Impact
O_a	O_b	*2.85*
O_a	O_c	*3.22*
O_a	O_h	*9.08*
O_a	O_g	*4.33*
O_a	O_f	*1.25*

In Table 10, we estimate the *Illegal Information Flow Risk*. For this calculation we combine the values of *Illegal Information Flow Impact*, and *Illegal Information Flow Likelihood*.

Table 10: Calculation of Illegal Information Flow Risk.

From	To	Illegal Information Flow Risk Value	Illegal Information Flow Risk Level
O_a	O_b	*1.43*	*Very Low*
O_a	O_c	*3.22*	*Low*
O_a	O_h	*9.08*	*Very High*
O_a	O_g	*4.33*	*Medium*
O_a	O_f	*0.00*	*Very Low*

We observe that the information flow from object A to object H contains a very high risk. This indicates that there is great possibility any transaction with such an information flow to be illegal. Thus, security experts should pay enough attention on objects A and H. The current access control rules should be examined and more accurate ones should be enforced.

6 CONCLUSIONS

In this work, we propose an approach to detect the illegal information flows in IIoT environments. For this purpose, we utilise a risk-based analysis method for assessing the information flows on bussiness processes. For each information flow we calculate the metrics of: (a) illegal information flow likelihood, and (b) illegal information flow risk. This will help to control the information that illegally transmitted from one node to another within an IIoT infrastructure network.

We demonstrate the applicability of our method by presenting an example that is composed of both legal and illegal information flows. Overall, our method can successfully identify the high-risk information flows, and thus the high-risk nodes that are susceptable to participate in an illegal information flow. This is an important feedback, because we can point out the nodes that are vulnerable and need more accurate access control rules.

Our future work is to identify whether potential illegal information flow risk is transferred from the previous connection to the next one, where illegal information flow of one transaction may be propagated to the next transaction within a business process. Our vision is to evaluate our approach in real world IIoT environments, such as smart manufacturing or smart grid.

ACKNOWLEDGEMENTS

This work has been supported in part by a research grant offered by the Hellenic Ministry of Digital Governance to the Research Centre of the Athens University of Economics & Business, Greece (2022-24).

REFERENCES

Bell, D., & La Padula, L., Secure computer systems: Unified exposition and Multics interpretation. MTR-2997, MITRE Corp., USA, 1976.

Cai, G., Shi, L., & Sui, X. (2009, October). Illegal Information Flow Detection in Electronic Institution. In *2009 3rd International Conference on Genetic and Evolutionary Computing* (pp. 240-243). IEEE.

Denning, D. E. (1976). A lattice model of secure information flow. *Com. of the ACM*, *19*(5), 236-243.

Downs, D., Rub, J., Kung, K., & Jordan, C. (1985, April). Issues in discretionary access control. In *1985 IEEE Symposium on Security and Privacy* (pp. 208-208). IEEE.

ENISA (2018). *Good Practices for Security of Internet of Things in the context of Smart Manufacturing*. European Union Agency for Network and Information Security (ENISA).

Hammer, C., & Snelting, G. (2009). Flow-sensitive, context -sensitive, and object-sensitive information flow control based on program dependence graphs. *International Journal of Information Security*, 8(6), 399-422.

Hedin, D., & Sabelfeld, A. (2012). A perspective on information-flow control. In *Software safety and security* (pp. 319-347). IOS Press.

Jaume, M., Tong, V., & Mé, L. (2011, December). Flow based interpretation of access control: Detection of illegal information flows. In *ICISS* (pp. 72-86).

Kotzanikolaou, P., Theoharidou, M., & Gritzalis, D. (2013 a, March). Cascading effects of common-cause failures in critical infrastructures. In *7th IFIP International Conference* (pp. 171-182). Springer.

Kotzanikolaou, P., Theoharidou, M., & Gritzalis, D. (2013 b). Assessing n-order dependencies between critical infrastructures. *International Journal of Critical Infrastructures 6*, *9*(1-2), 93-110.

Masri, W., Podgurski, A., & Leon, D. (2004, November). Detecting and debugging insecure information flows. In *15th International Symposium on Software Reliability Engineering* (pp. 198-209). IEEE.

Nakamura, S., Ogiela, L., Enokido, T., & Takizawa, M. (2018). An information flow control model in a topic-based publish/subscribe system. *Journal of High-Speed Networks*, *24*(3), 243-257.

Nakamura, S., Enokido, T., Barolli, L., & Takizawa, M. (2019, June). Capability-based information flow control model in the IoT. In *13th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing* (pp. 63-71). Springer

Salonikias, S., Gouglidis, A., Mavridis, I., & Gritzalis, D. (2019). Access control in the industrial internet of things. *Security and privacy trends in the industrial internet of things*, 95-114.

Samarati, P., & de Vimercati, S. C. (2001). Access control: Policies, models, and mechanisms. In *Foundations of Security Analysis and Design: Tutorial Lectures 1* (pp. 137-196). Springer.

Sandhu, R. (1993). Lattice-based access control models. *Computer*, *26*(11), 9-19.

Sandhu, R., Coyne, E., Feinstein, H., & Youman, C. (1996). Role-based access control models. *Computer*, 29 (2), 38-47.

Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. *IEEE transactions on industrial informatics*, *14*(11), 4724-4734.

Zimmermann, J., Mé, L., & Bidan, C. (2003). An improved reference flow control model for policy-based intrusion detection. *Lecture notes in Computer Science*, *2808*, 291-308.