Enabling fine-grained Access Control in Information Sharing with Structured data Formats

Tatu Niskanen and Jarno Salonen
VTT Technical Research Centre of Finland, Espoo, Finland

tatu.niskanen@vtt.fi
jarno.salonen@vtt.fi

Abstract: The ongoing need for societal and industrial digital transformation requires rapidly expanding networks of interconnected organizations and dictates an increasing role for cybersecurity in information sharing. A typical setup consists of multiple stakeholders working closely together and needing efficient channels for sharing relevant information in a secure manner. This is especially prevalent with complex modern supply chains and critical information infrastructures. They often comprise of numerous co-operating organizations, people and in some cases smart devices having different levels of access to a variety of information. Granular access control plays a vital role when distributing information efficiently between stakeholders without revealing sensitive pieces of data to unwanted third parties. This article presents a novel framework for enabling fine-grained access control to share information efficiently and securely in these situations. Our motivation and use case for the framework originates from the secure sharing of cyber incident information in the maritime logistics industry. We present a novel solution to this problem by developing an information sharing platform and a meta-model, demonstrated using an implementation with structured JSON data formats, while supporting previously researched attribute-based encryption schemes. The proposed framework provides a broader context to the fine-grained data access control challenge in addition to the technical implementation.

Keywords: Cybersecurity, Information security, Access control, Information sharing, Incident management, Resilience.

1. Introduction

Widespread and complex supply chains are common in the modern world of industry 4.0. This is evident for example in the automotive industry, where a single product has countless parts with separate manufacturers embedded into it. The supply chains and stakeholder networks surrounding these products are complex and require well refined measures to share all the necessary information securely with everyone involved. This is also evident with critical information infrastructures (CIIs). CIIs usually have complex architectures composed of many interconnected components with personnel having access to different levels of information. This article presents a novel framework to enable fine-grained access control (FGAC) for secure and efficient information sharing in situations like these ones.

FGAC systems facilitate granting differential access rights to a set of users and allow flexibility in specifying the access rights of individual users (Goyal et al, 2006). FGAC allows these access rights to be implemented using higher granularity than what traditional folder or file-level access control makes possible. This becomes relevant when we need to share a bundle of information, like a manual or a log file securely and efficiently with many stakeholders, while considering the different authorization of each individual to the specific data points. With a well-refined FGAC solution, we can share the entire manual or log file as such to the stakeholders and control read and write access to the different sections for everyone separately.

We produced a practical use case for our framework from sharing cyber incident information among stakeholders in the maritime logistics industry and related supply chains. An increasing number of complex cyber-attacks has affected the maritime sector, which is somewhat typical for modern industries relying heavily on cyber-connected systems. In order to combat this, Jacq et al. (2021) have identified a need for increasing the cyber risk awareness of the industry's smart ports.

In order to achieve this vision of increased cyber risk awareness, it is vital to define appropriate mechanisms for sharing and managing of threat and incident information. With our proposed solution, organizations can share incident data quickly and efficiently, while allowing access to the individual data points only to authorized actors. We validate our framework in a case-study implementation to a maritime cybersecurity exercise environment. Although our model allows for FGAC in sharing of cyber incident information, it is not limited to this type of information. We can extend our meta-model to most other types of information one wishes to share in their operational environment.

Our research contribution is a practical framework that organizations can implement to enable FGAC in information sharing. It utilizes a novel and flexible solution that can be applied to any data format which uses name/value pairs. We validate the framework with a case-study. During our research, we also discovered that

the extant literature has to some degree been very centred around the technical implementation and has thus partially left out the surrounding context regarding implementing FGAC in real-life environments. Therefore, in addition to the novel technical solution, our research also addresses some important issues regarding this surrounding context. Our research questions are the following:

1. How can we enable FGAC in sharing of sensitive information?
2. How can we create structured data formats which enable granular encryption and decryption?
3. How can we create a practically applicable framework which organizations can use to enable FGAC in information sharing?

We have structured the article as follows. After a brief introduction to the topic in this chapter, we describe the relevant theoretical background in chapter two. In chapter three, we present our framework and the four processes that enable its' implementation. In chapter four, we implement the framework in a case-study targeted to the maritime environment. Finally, we discuss about the implementation and conclude the article in chapter five.

2. Theoretical background

In this section we present the most noteworthy findings from current literature and research regarding our application of FGAC. We discuss about alternative FGAC implementations, relevant access control and encryption schemes, and some previous research concerning information sharing with structured data formats.

Access control is the process of mediating every request to resources and data maintained by a system and determining whether the request should be granted or denied (Samarati et al, 2000). Different access control policies can be applied, which correspond to different criteria on what is allowed. These policies form a basis for many different access control models, which represent how an organization enforces access control in practice, for example in a computer system. The different access control models are a well-researched domain and understanding them forms the basis for applying FGAC in information sharing. Some of the most implemented and researched access control models include role-based access control (Sandhu, 1998), attribute-based access control (Federal CIO Council, 2009 and Hu et al, 2015), and organization-based access control (Kalam et al., 2003). We further elaborate these models in the next section, alongside discussing the appropriate underlying access control models for applying FGAC.

The enforcement of access control often depends on the underlying encryption schemes. Some schemes are more suitable than others, or even specifically designed to carry out FGAC. An example of an encryption scheme designed to achieve FGAC is the work done by Goyal et al (2006). This encryption scheme called attribute-based encryption (ABE) works by labelling ciphertexts with a set of attributes. Cryptographic decryption keys are then associated with access structures that control which ciphertexts the user can decrypt. This approach however has some drawbacks, mainly related to the lack of a straightforward attribute and key revocation mechanisms.

FGAC has been the subject of research in different contexts. Some research has focused on the application of FGAC in the cloud environment. The work by Wang et al. (2010) presents a solution to achieve FGAC with the previously mentioned ABE schemes in the cloud environment. FGAC has also been implemented using blockchain solutions. Wang et al. (2018) present a blockchain-based framework to achieve FGAC and mention that the private key generator has the ability to decrypt all data stored in the cloud server in all ABE schemes, which may result in serious problems such as key abuse and privacy data leakage. They aim to provide a blockchain-based solution, which solves the problem of a single point of failure with a decentralized storage model. Blockchain however also has some disadvantages, and it may not always be the preferred solution by the implementer. The disadvantages of blockchain implementations include for example high transaction costs and difficult integration especially with older systems. (Golosova et al, 2018.)

With regards to the act of sharing information, especially considering our motivation of applying FGAC to the sharing of cyber incident information, some key background can be found in the RFC-standards. RFC 5070 defines the incident object description exchange format (IODEF). This is a standardized format which is used to share computer security information between computer security incident response teams. (Danyliw et al, 2007.) IODEF version 2 specified in RFC 7970 introduces new information points to this format (Danyliw, 2016). Salonen et al. (2022) have introduced a somewhat primitive model to enhance the sharing of cyber incident information via FGAC, utilizing IODEF to achieve this goal. This article provides a more mature version of that model and extends it beyond IODEF and incident information sharing.

3. Designing the Framework

We designed a framework to achieve FGAC for granular information sharing, with the purpose of presenting a replicable solution to this problem. We constructed the framework according to the following process:

1. Establishing and verifying user roles/attributes.
2. Setting up an information sharing channel.
3. Establishing a structured format for the information type.
4. Distributing access rights to the shared information.

In this section, we explain these parts and describe them in further detail. The first step makes the future work less demanding by providing a clear picture of the users' access control requirements in the implementer's operational environment. The second part is about setting up the venue through which we handle information sharing and apply FGAC. In the third part, a structured format is established for a given information type. This predefined format is needed to make the FGAC solutions of this framework implementable. The fourth section handles the software solutions, which allow for distribution of access rights, and the encryption methods.

3.1 Establishing and verifying user roles/attributes

In order to succeed in applying FGAC securely and efficiently, we must first plan and verify appropriate user roles or attributes in the access control framework of the target environment. This is vital for formulating a clear picture of who requires access to which information and becomes critical later when granular access rights are distributed to the shared information. The proper implementation of this step causes less manual work in later steps.

Multi-organization environments are the most relevant targets for FGAC. Establishing a solid access control policy and managing all the individual access rights can be challenging when multiple organizations operate concurrently with differing systems, personnel, and access control models. Some access control models are specifically designed to make handling this task more manageable, such as the Multi-OrBAC (Abou El Kalam et al, 2006) and PolyOrBAC (Abou El Kalam et al, 2009) models. They build on the foundations of organization-based access control (OrBAC) (Kalam et al, 2003). These applications are worth exploring, and the chosen access control model allows a freedom of choice for the implementer. However, for the purposes of applying this framework, a role-based or attribute-based solution is sufficient. The priority is that the chosen model gives us a clear picture of the different roles or attributes that the users have, which we can base our access control on.

Role-based access control (RBAC) is a policy-neutral model based on roles and role permissions (Sandhu, 1998). RBAC is known to have some pitfalls, which should be acknowledged. RBAC is not always well suited for provisioning all access control happening in a complex organization, since this can quickly lead to an explosion in the number of roles. For this reason, it should only be applied in applications which it is well suited for. In the more complex cases, RBAC may prove incapable of specifying users' access rights, since specifying access control based on the user's role alone may prove inadequate. It is also important to understand that RBAC is an ongoing effort. As organizations change, role models must be reviewed and updated to match. Even though RBAC is known to have its limitations, it is still a valid solution in certain applications. (Ferraiolo et al, 2001)

Attribute-based access control (ABAC) is a model where a subject's authorization to perform a set of operations is based on attributes associated with the subject, object, requested operations, and in some cases environment attributes (Hu et al, 2015). It is sometimes seen as the successor to RBAC. This framework is well suited for an ABAC implementation, although we demonstrate it using only the role-attribute in our case-study. An ABAC implementation does allow for further specifying of access rights using other attributes. We elaborate on this in the fourth step where we discuss the technical implementation.

3.2 Setting up an information sharing channel

In order to handle the sharing of documents, along with enabling their granular encryption and decryption, we must establish an information sharing channel. This makes implementing FGAC manageable in the scope of this framework. The channel acts as an intermediary between the information distributor and the consumer. It can be set up using a software-as-a-service implementation. Using this service, we can implement the software/encryption solutions required to apply FGAC to the shared information. We also use this service to view and manage the information, although this could be conducted locally by the users. The service needs to allow for the storage of:

- Encrypted documents
- User data to manage authorization
- Cryptographic keys for appropriate decryption of information

It is largely up to the implementer how they integrate user data into the service, as long as the users signing in have appropriate roles or attributes for managing access control. Figure 1 illustrates a possible service architecture.

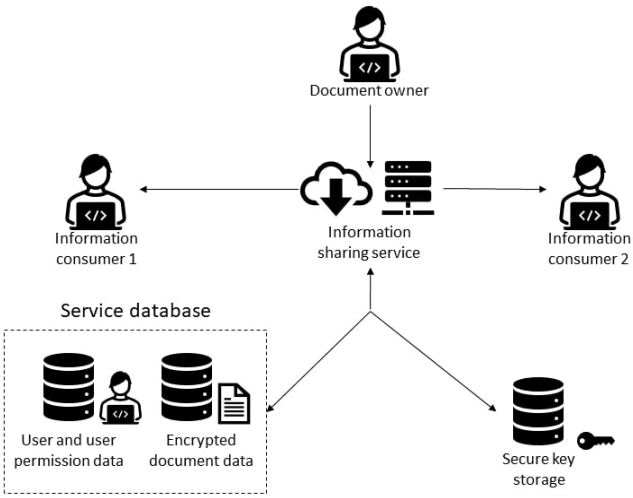

Figure 1 Illustration of a possible information sharing service architecture

The data should be stored in a secure database in an encrypted format. Our proposed solution is a MongoDB-database due to the flexibility its' NoSQL structure offers for managing JSON data. Depending on the chosen encryption scheme, the decryption will either be performed with securely stored document keys by the service, or the users' private keys using an ABE solution. We return to the different options for encryption in the fourth step.

The cryptographic keys should always be stored in a separate location from the encrypted data, as recommended by the Open Web Application Security Project (OWASP, 2022). This can mean storing them separately in the same system, but it is worthwhile to consider storing the keys on a separate system altogether. It is a known fact that the secure storage of cryptographic keys is a difficult problem to solve when the application needs to have some level of access to the keys in order to decrypt the data. OWASP recommends additional secure storage mechanisms for further security. These include a physical or a virtual hardware security module (HSM), key vaults, and secure storage APIs. (OWASP, 2022)

3.3 Establishing a structured format for the information type

In this part, a structured data format is created for the information to be shared. We use it to implement the encryption solutions which enable FGAC. The organization or multi-organization environment may want to come to agreement on a standardized format for the information type when possible. This eases the creation of subsequent documents of the same type.

The recommended data format in this framework is the JSON data format. JSON supports multiple individual fields and sub-fields using name/value pairs, which suits the methods of applying FGAC in this framework. The syntax and hierarchical structure of JSON strings make them easily interpretable by applications, and JSON can be parsed into a ready-to-use JavaScript object. As simple text, JSON is also suitable and safe for transferring across platforms and operating systems that do not share more complex document types. It is a format designed specifically for data interchange, outperforming alternatives such as XML in terms of speed and memory consumption. (Zunke et al, 2014)

For demonstration purposes, we assume that the shared information is the operations manual for a system component in a manufacturing environment. We can then construct a structured JSON format for that manual, resembling the IODEF data format. Table 1 clarifies this with an example. We apply this same method to our incident information sharing use case.

Table 1 Structured data format for a component manual

Field	Multiplicity	Authorization	Description
ComponentID	One	A4, B2, C6	A component identification number assigned to this component by the document creator
PartList	One	A4, C6	List of the parts contained in the component
ComponentDescription	One	B2	Description of the component
AssemblyManual	One	B2, C6	Manual to assemble the component
Contact	One or more	A4, B2, C6	Contact information to inquire further information
AdditionalData	Zero or one	A4, B2	Additional relevant data about the component

The table presents an example of a predetermined and structured data format. The authorization tags can correspond to user roles such as "maintenance worker", or other user attributes, which are used for access control management. Writing these attributes as corresponding tags instead of clear text descriptions adds an extra layer of security alongside a simpler presentation. The fields and possible sub-fields should clearly hold one type of information with a clear and concise authorization classification. When utilizing JSON, this format can be supported with a JSON-schema.

3.4 Distributing access rights to the shared information

This workflow step handles the embedding of access rights into the shared documents, along with the encryption methods and software solutions used in order to successfully manage FGAC.

After we have defined the authorization levels of the individual fields, the encryption process is performed. Enabling FGAC can be achieved by distributing one or more identifiers called access control tags to the individual fields. These correspond to user roles or other user attributes. We base these tags on user roles in our implementation. Access control tags remain as plaintext when we encrypt the document and bring it to the database, which allows for easier verification of privileges. The JSON data format does not support inserting images directly, but there are workarounds for this. One possible solution is to encode the images as ASCII text using Base64 encoding. Figure 2 displays the embedding of access control tags within a JSON file.

```
"ComponentId": {
  "roleTags": [
  "a4",
  "b3",
  "c4"
  ],
  "content": "f474a"
},
"PartList": {
  "roleTags": [
  "a4",
  "c2"
  ],
  "content": "Part 1, part 2, part 3."
},
```

Figure 2 Distributed access control tags in a JSON file

The distributed role-tags in this example correspond to the user roles which require access to the encrypted content within the field. All users registered to the service hold one or more roles. For example in a manufacturing environment, the distributed role-tag "a4" can correspond to a maintenance worker in a certain factory, whose role is indicated by a corresponding attribute. The permission to access/decrypt this field is then granted to the users of this role. With an ABAC solution it is also possible to implement tags here that correspond to other attributes besides roles. Figure 3 demonstrates an ABAC implementation:

```
"ComponentId": {
  "projectTags": [
  "a1",
  "b2",
  "c5"
  ],
  "locationTags": [
  "f7",
  "g3"
  ],
  "content": "f474a"
},
```

Figure 3 ABAC implementation of access control tags

We identify two suitable methods to handle the encryption using this solution:

- Symmetric-key server-side encryption and decryption.
- Attribute-based encryption schemes.

With the symmetric-key option, the implemented software solution is responsible for appropriate decryption of information for the users. We can check the access control tags under the JSON-fields directly against user roles or other attributes during decryption requests. Figure 4 demonstrates this in JavaScript.

```
if requestingUserTags.some(r =>
documentContent[field].tags.includes(r))
```

Figure 4 Verifying privileges in a JavaScript implementation

The encrypted information within the field can be decrypted for the user by the server if this requirement is fulfilled. The server fetches a corresponding document key for decryption from a secure key storage.

We base the second method of handling the encryption on previous work done by Goyal et al (2006). Applying ABE with our model requires that the underlying software solution labels ciphertexts with the appropriate attributes during the encryption/upload process, based on the access control tags implemented under the JSON fields. The software then generates corresponding private keys and distributes them to the users, enabling an asymmetric key scheme. The demonstration of this technical implementation will be future research.

4. Implementing the framework: a case-study

We developed a proof-of-concept information sharing service that demonstrates the framework in a practical setting. The use-case is based on the secure sharing of host-based and network monitoring data in maritime environment cybersecurity exercises. During Capture-the-Flag (CTF) type incident response exercises, we share network monitoring data captured during a simulated incident to the participants. The participants are granted access to the monitoring data according to their role and privileges in the context of the exercise. Thus, some of the participants are initially given limited access to the monitoring data, with only certain data points accessible. The data is used to formulate a situational picture of what has happened in the incident. The participants can be granted access to additional data points within the monitoring data, if they are initially unable to gain an adequate understanding of the situation.

The implementation of the service utilizes the methodologies presented in this framework. *We initiate the information sharing process of our case-study according to the first workflow step by defining what roles the participants have within the exercise*. We define two distinct roles; some participants are given managerial roles with more privileges, while others are left with basic roles with more limited access to the data. *As the second step of our framework dictates, we set up an information sharing channel to enable FGAC in the sharing of data to the participants*. Figure 5 illustrates the implemented information sharing service architecture in this case study:

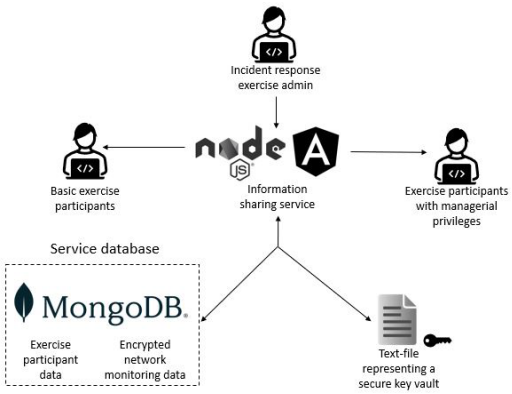

Figure 5 Implemented service architecture

The platform front-end was developed with the Angular 2+ web application framework for TypeScript, and the back-end solution was built with JavaScript's Node.JS framework. The tag-based encryption and the document-specific key generation is handled by the back-end solution, while the participants use the front-end interface to view the incident data during the exercise. We save the encrypted monitoring data and exercise participant data to a MongoDB-database, while we use a basic text file to save the keys. Although a text file in this situation is able to represent a secure key storage, we prefer other solutions like proper secure key vaults with a HSM in a real-world setting. The encryption is performed with the Advanced Encryption Standard using JavaScript's CryptoJS-library.

As instructed by the third step of our framework, we establish a structured format for the shared information. The network monitoring data for the exercise is polled with the Elasticsearch search engine. Elasticsearch is a suitable tool for our model since it can produce a JSON file of the entire network monitoring data, which can be uploaded to the service. Therefore, in this case we can produce our structured data format using the Elasticsearch search engine in a ready-to-use JSON-format. As the Elasticsearch query determines the fields and their content, we only need to note it down and define the authorization levels of each field for the users. With the roles defined in the first step, we use the indicator "u1" for basic users and "u2" for users with heightened privileges.

As the final step of our framework, *we distribute access rights to the data*. As an intermediary operation, we input the appropriate access control tags under the fields of the JSON file before the exercise admin uploads it to the service. We conduct this process manually in this case, but we could implement automated software solutions for this step in a relatively straightforward fashion. The encryption process is initiated when the document owner, in this case the exercise administrator, uploads the JSON file to the service. Regarding the chosen approach to encryption, we chose the symmetric-key option instead of ABE schemes. The software encrypts the field contents, while the tags and title remain as cleartext. The encrypted file is then saved to the MongoDB-database, as partially shown in Figure 6. A corresponding document key is saved to a secure key storage.


```
_id: 1
▾ fields: Array
  ▾ 0: Object
      content: "U2FsdGVkX18U0J5CU+AdXknA971Yhw1a56V2qIe3wsk="
    ▾ tags: Array
        0: "u1"
        1: "u2"
      title: "Incident-ID"
  ▾ 1: Object
      content: "U2FsdGVkX1+7E3OccuBGuzlytl7aSRCc5YstTqkqcnCGH7tDZ2y0ngUWrVcvUTbIYcFLwS…"
    ▾ tags: Array
        0: "u1"
        1: "u2"
      title: "Description"
  ▾ 2: Object
      content: "U2FsdGVkX1/BGGUNp1EIj27+A68UUYL/fstjbvkp7n8KclIsprYJismer3nXSwRj"
    ▾ tags: Array
        0: "u2"
      title: "Agent-ID"
```


Figure 6 JSON file parsed and encrypted into the database

With all the steps performed, we are able to display the implemented front-end for the case study users as our end result in Figure 7. It shows how a user is logged in as a basic participant with limited access to the data points, dictated by the implemented FGAC solution. When exercise participants access the service, they see a list of all the files that include tags which correspond to them. The decryption process is initiated when participants request to view a file by clicking it. The software goes through the document fields and decrypts data based on the access control tags that match the participant attributes within their user data.

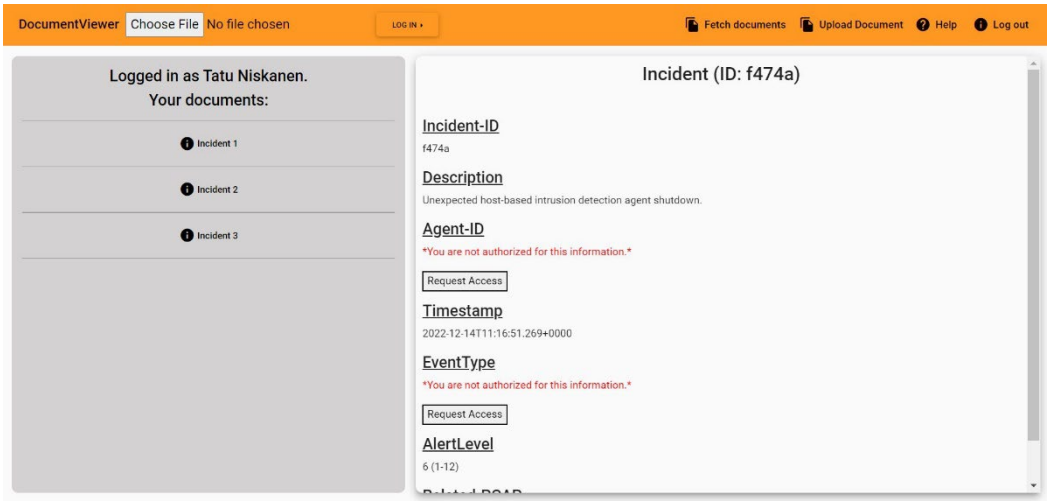

Figure 7 View of the front-end implementation with limited privileges

5. Discussion and conclusion

This article presents a practical solution to applying fine-grained access control (FGAC) in information sharing. The solution is based on a novel framework, motivated by information security needs identified within the maritime logistics and manufacturing industries. We have enabled it by using structured data formats and adding data specific access control tags to the individual data contents, while providing a broader context around the technical implementation.

By adopting the framework, organizations can enable FGAC in sensitive information sharing. The framework consists of four steps ranging from the establishment of user roles/attributes and setting up a suitable information-sharing channel, to establishing a structured data format and distributing access rights to the shared information. We describe a case study to demonstrate our framework and the practical use of FGAC in information sharing. In the case of utilizing this framework within another use case, the first workflow step needs to be implemented to fit the specific needs of the operational environment. The second step may be used as presented, although it can be altered to suit additional technical requirements. The third and fourth steps are in general re-usable in the presented form.

We provide an example on the establishment of a structured data format that enables granular encryption and decryption. Our example is based on the JSON data format, but the framework is compatible with any data format that supports name/value pairs. The framework supports multiple different encryption methods. Although we demonstrated the utility of this framework with the user roles as the determining attributes, other attributes could be used in their place. This can mean for example projects and locations that the users are assigned to.

Our future research and development activities consist of a more comprehensive piloting and evaluation of the service that can be done, e.g., in some other suitable future project. This would consist of among others a trial with actual users and real incident data that is added into the service and used for analysis. During the pilot, we could evaluate the service efficiency and applicability, as well as conduct surveys to collect end-users' views concerning the advantages and disadvantages of the service and system compared to other existing services and tools focused on incident management. Our future work also includes thorough security testing of the presented method. Another interesting topic for piloting would be evaluating the use and feasibility of alternative encryption mechanisms and/or data formats.

According to our research, the extant literature on FGAC has to some degree been mainly focused on the technical implementation and not so much on the surrounding context such as viable business cases. Therefore, we propose that more research should be conducted on the non-technical perspective and especially in realistic use cases such as the Cyber-MAR project (Jacq et al, 2021). Our proposed framework provides a template on how organizations can enable FGAC in information sharing. The future objective is to further validate the framework using real scenarios and to compare it more thoroughly with other potential competitors, adopting best practices and learning from their weaknesses.

Acknowledgements

This article is based on research and development work conducted in two Horizon 2020 projects, namely Secure Collaborative Intelligent Industrial Assets (SeCoIIA) and Cyber preparedness actions for a holistic approach and awareness raising in the MARitime logistics supply chain (Cyber-MAR). SeCoIIA aims at securing the digital transition of manufacturing industry towards more connected, collaborative, flexible and automated production techniques. Cyber-MAR aims to develop an innovative cybersecurity simulation environment for accommodating the peculiarities of the maritime sector, while being easily applicable in other transport subsectors, with the view to fully unlock the value of the use of cyber range in the maritime logistics value chain. The projects have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871967 and No. 833389.

References

Abou El Kalam, A. and Deswarte, Y. (2006). Multi-orbac: A new access control model for distributed, heterogeneous and collaborative systems. In Proceedings of the IEEE Symposium on Systems and Information Security.

Abou El Kalam, A., Deswarte, Y., Baïna, A., and Kaâniche, M. (2009). Polyorbac: A security framework for crit- ical infrastructures. International Journal of Critical Infrastructure Protection, 2(4):154–169.

Danyliw, R. (2016). The incident object description exchange format version 2. RFC 7970, RFC Editor.

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. (2001). Proposed nist standard for role-based access control. ACM Transactions on In- formation and System Security (TISSEC), 4(3):224– 274.

Golosova, J. and Romanovs, A. (2018). The advantages and disadvantages of the blockchain technology. In 2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE), pages 1–6. IEEE.

Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security, pages 89–98.

Federal CIO Council (2009) Federal Identity, Credential, and Access Management (FICAM) Roadmap and 1671 Implementation Guidance Version 1.0, November 10, 2009.

Hu, V. C., Kuhn, D. R., Ferraiolo, D. F., and Voas, J. (2015). Attribute-based access control. Computer, 48(2):85– 88.

Jacq, O., Salazar, P. G., Parasuraman, K., Kuusijärvi, J., Gkaniatsou, A., Latsa, E., and Amditis, A. (2021). The cyber-mar project: First results and perspectives on the use of hybrid cyber ranges for port cyber risk assessment. In 2021 IEEE International Conference on Cyber Security and Resilience (CSR), pages 409– 414. IEEE.

Kalam, A. A. E., Baida, R. E., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miege, A., Saurel, C., and Trouessin, G. (2003). Organization based access con- trol. In Proceedings POLICY 2003. IEEE 4th Interna- tional Workshop on Policies for Distributed Systems and Networks, pages 120–131. IEEE.

OWASP (2022). Owasp cheatsheets. https: //cheatsheetseries.owasp.org/cheatsheets/Cryptographic Storage Cheat Sheet.html. Accessed: 2022-08-15.

R. Danyliw, J. Meijer, Y. D. (2007). The incident object description exchange format. RFC 5070, RFC Editor.

Salonen, J., Niskanen, T., and Raitio, P. (2022). How to enhance the sharing of cyber incident information via fine-grained access control. In 3rd International Conference on Data Mining and Machine Learning, DMML 2022, page 17. AIRCC Publishing Corporation.

Samarati, P. and Vimercati, S. C. d. (2000). Access control: Policies, models, and mechanisms. In International School on Foundations of Security Analysis and De- sign, pages 137–196. Springer.

Sandhu, R. S. (1998). Role-based access control. In Advances in computers, volume 46, pages 237–286. Elsevier.

Wang, G., Liu, Q., and Wu, J. (2010). Hierarchical attribute- based encryption for fine-grained access control in cloud storage services. In Proceedings of the 17th ACM conference on Computer and communications security, pages 735–737.

Wang, S., Zhang, Y., and Zhang, Y. (2018). A blockchain- based framework for data sharing with fine-grained access control in decentralized storage systems. Ieee Access, 6:38437–38450.

Zunke, S. and D'Souza, V. (2014). Json vs xml: A comparative performance analysis of data exchange formats. IJCSN International Journal of Computer Science and Network, 3(4):257–261.