

This work was written as part of one of the author's official duties as an Employee of the United States Government and is therefore a work of the United States Government. In accordance with 17 U.S.C. 105, no copyright protection is available for such works under U.S. Law.

Public Domain Mark 1.0

<https://creativecommons.org/publicdomain/mark/1.0/>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

**Please provide feedback**

Please support the ScholarWorks@UMBC repository by emailing [scholarworks-group@umbc.edu](mailto:scholarworks-group@umbc.edu) and telling us what having access to this work means to you and why it's important to you. Thank you.

# MATS: A Multi-aspect and Adaptive Trust-based Situation-aware Access Control Framework for Federated Data-as-a-Service Systems

Dae-young Kim\*, Nujood Alodadi\*, Zhiyuan Chen\*, Karuna P. Joshi\*, Adina Crainiceanu†, Don Needham†

\*University of Maryland, Baltimore County, Baltimore, MD 21250 USA

{leroy.kim, nujood1, zhchen, karuna.joshi}@umbc.edu

†United States Naval Academy, Annapolis, MD 21402 USA

{adina, needham}@usna.edu

**Abstract**—Federated Data-as-a-Service systems are helpful in applications that require dynamic coordination of multiple organizations, such as maritime search and rescue, disaster relief, or contact tracing of an infectious disease. In such systems it is often the case that users cannot be wholly trusted, and access control conditions need to take the level of trust into account. Most existing work on trust-based access control in web services focuses on a single aspect of trust, like user credentials, but trust often has multiple aspects such as users’ behavior and their organization. In addition, most existing solutions use a fixed threshold to determine whether a user’s trust is sufficient, ignoring the dynamic situation where the trade-off between benefits and risks of granting access should be considered. We have developed a Multi-aspect and Adaptive Trust-based Situation-aware Access Control Framework we call “MATS” for federated data sharing systems. Our framework is built using Semantic Web technologies and uses game theory to adjust a system’s access decisions based on dynamic situations. We use query rewriting to implement this framework and optimize the system’s performance by carefully balancing efficiency and simplicity. In this paper we present this framework in detail, including experimental results that validate the feasibility of our approach.

**Index Terms**—trust, access control, semantic web, game theory

## I. INTRODUCTION

Complex applications that require dynamic coordination of multiple organizations, such as maritime search and rescue (SAR) operations, disaster relief, or contact tracing of an infectious disease, can benefit from the use of federated Data-as-a-Service systems. Such systems consist of multiple members who own their private data but would like to exchange information necessary for a common mission. For example, a vessel in distress wants to share its location and direction of movement with other vessels in a SAR mission. Each member of the federated system has data that needs to be kept private, as well as data that needs to be shared with other collaborators to accomplish the mission. Current approaches to data sharing are centered around situation-aware access control, also called policy-based or attribute-based access control [1]–[3]. Members may join the mission at any time, and data access decisions depend on situations like whether a ship is in distress or a person tests positive for a disease.

However, in such systems, we often cannot completely trust a user in this dynamic network as there may be malicious users or those who abuse their privileges. For example, the U.S. Department of Justice [4] reported many types of fraud in disaster relief, including identity theft, fraudulent solicitations for donations and charitable giving, insurance claim fraud, etc. It is estimated that improper and potentially fraudulent individual assistance payments are between \$600 million and \$1.4 billion in Hurricanes Katrina and Rita disaster relief [5]. Due to these factors, it is clear that access control mechanisms need to take the level of trust into account. Existing work on trust-based access control in web services [6] often focuses on just a single aspect of trust, such as user’s credentials, but trust often has multiple aspects such as users’ behavior over time or the organization to which the user belongs. In addition, most such solutions use a fixed threshold to determine whether a user’s trust is sufficient, ignoring the dynamic situation where the trade-off between the benefits and risks of granting access should be considered. Setting the threshold too high may deny legitimate users, while setting it too low may give access to malicious users.

One possible solution to address this issue is to use game theory [7]. However, integrating game theory and maintaining trust in a federated system adds to complexity of the system and incurs extra cost. For example, the system needs to decide where to store the information related to trust, and reduce the communication overhead as in some use cases, such as maritime SAR, network bandwidth is limited.

We propose *MATS*, a novel Multi-aspect and Adaptive Trust-based Situation-aware Access Control Framework for federated data sharing systems that addresses the technical challenges identified above. Our framework is built using Semantic Web technologies and uses game theory to adjust the system’s decisions based on dynamic situations. Key contributions of the *MATS* framework include:

- 1) A multi-aspect trust-based access control framework that integrates identity trust, behaviour trust, organization trust, and user roles;
- 2) A situation-aware access control framework that integrates a game theory model to dynamically update trust

and the system's access control decisions;

- 3) An efficient approach to implementing the proposed framework in a federated semantic-web based architecture by using query rewriting. This allows access control checking to be part of query evaluation and uses existing semantic-web frameworks without the need to implement additional complex modules such as distributed reasoning.

The remainder of the paper is organized as follows. Section II discusses related work. Section III describes our approach. Experimental results are presented in Section IV and Section V gives our conclusions.

## II. RELATED WORK

There is limited previous research on SAR mission ontologies. Weihong designed a maritime search and rescue decision-making ontology to solve semantic heterogeneity of information for SAR missions [8]. The ontology includes Event, Ship, Incident Level, Volume of Oil Spill, and Person classes and their subclasses to represent information about the SAR mission. However, the ontology is limited to information about an incident itself, without classes regarding organizational collaboration, and gives the impression that the ontology is specific to oil spill accidents. Also, Weihong and Ruixin developed a marine search and rescue environment ontology to share knowledge about the search and rescue area and establish a knowledge database [9]. However, this ontology has the limitation that it only represents a specific portion of SAR mission data - the environment. Oni et al. [3] developed a SAR mission and Contact Tracing ontology to show real-world applications of situation-aware access control in federated systems. The SAR ontology included classes such as the vessel in distress, organizations participating in the search and rescue mission, assets owned by these organizations, the rescue coordination center, and their data in the system. It also incorporated roles, users, allowed operations (read or write), rescue missions in which users or organizations are involved, tasks in the mission, and others. A more sophisticated Contact Tracing ontology was proposed in [10]–[12] which contains slightly different classes such as electronic health records (EHR), travel history, and healthcare-related organizations and shared a similar structure with SAR ontology. However, all these works only considered behavioral trust based on policy compliance for access control. This paper introduces the game theory element to capture more complicated human behavior during the federated data exchange.

Yau et al. [1] propose a situation-aware access control framework for distributed settings with a model for representing access control rules. However, trust is not considered in their work.

There has been work on using semantic web technologies to enforce access control or privacy preferences. Beimeel and Peleg [2] propose a situation-aware access control model based on OWL ontology and SWRL rules. A similar semantic-based approach was proposed by Sun et al. [13] and applied to e-Healthcare. Kayes et al. [14] use an ontology-based solution

to represent purpose-oriented situations and use that in access control of software services. Oulmakhzoune et al. [15] use ontologies and query rewriting to enforce privacy preferences for data stored at a single place. Padia et al. [16] applies a query rewriting approach to enforce fine-grained access control to RDF data stored at a single place. Oni et al. [3] proposes a query rewriting method to enforce situation-aware access control in federated systems. However, none of these works consider trust.

The concept of trust has been used in access control systems in previous work. Researchers in [17] propose a model that extends traditional rule-based access control by incorporating user trust levels. Trust and context are both considered for assigning access rules based on context information and trustworthiness of users, which is evaluated using the reputation of a user among other users [18]. However, these models do not consider multiple factors that can affect trust.

Bahatti et al. [6] propose a trust-based and situation-aware access control framework for web services. Their solution treats trust as part of authorization and uses trust credentials to decide trust levels and roles for unknown users. Their framework also proposes dynamic adjustment of users' trust levels based on users' behavior as well as credentials. Bernal et al. [19] proposes a trust model for Internet of Things where trust is computed based on reputation, quality of service, security considerations and devices' social relationships. However, although both works allow users' trust levels to be adjusted, the threshold of required trust levels to access data is fixed, and does not consider associated risks and benefits.

Game theory allows a system to weigh risks and benefits of granting access to a user, and has been used to set thresholds for trust levels [7], [20]. He et al. uses game theory to analyze access control in a cloud environments where players include users and cloud service providers [20]. However their utility functions are abstract and do not provide details of how to quantify risks. Helil et al. proposes a non-zero sum game theory model [7] for access control. This model considers trust, risks, and cost in the utility functions. However it focuses on game theory in a client/server environment and does not discuss how to integrate it with the rest of an access control framework. Our paper integrates this model in a federated situation-aware access control framework and evaluates the framework's performance in a simulation study.

Game theory has been widely used in network security [21]. In particular, game theory is used to analyze the strategic behavior of users in access control systems in social networks [22]. Additionally, game analysis is used to model the interaction between users and the service provider in access control systems [7], [23]. Both studies use trust-based access control models to capture the dynamic behavior of users. The main goal is to provide different access permissions based on the users' trust levels so that users with high trust can access more resources. However, these models are mostly focused on centralized environments where users interact directly with the service providers. In our work, we consider a distributed environment where members can directly query data from

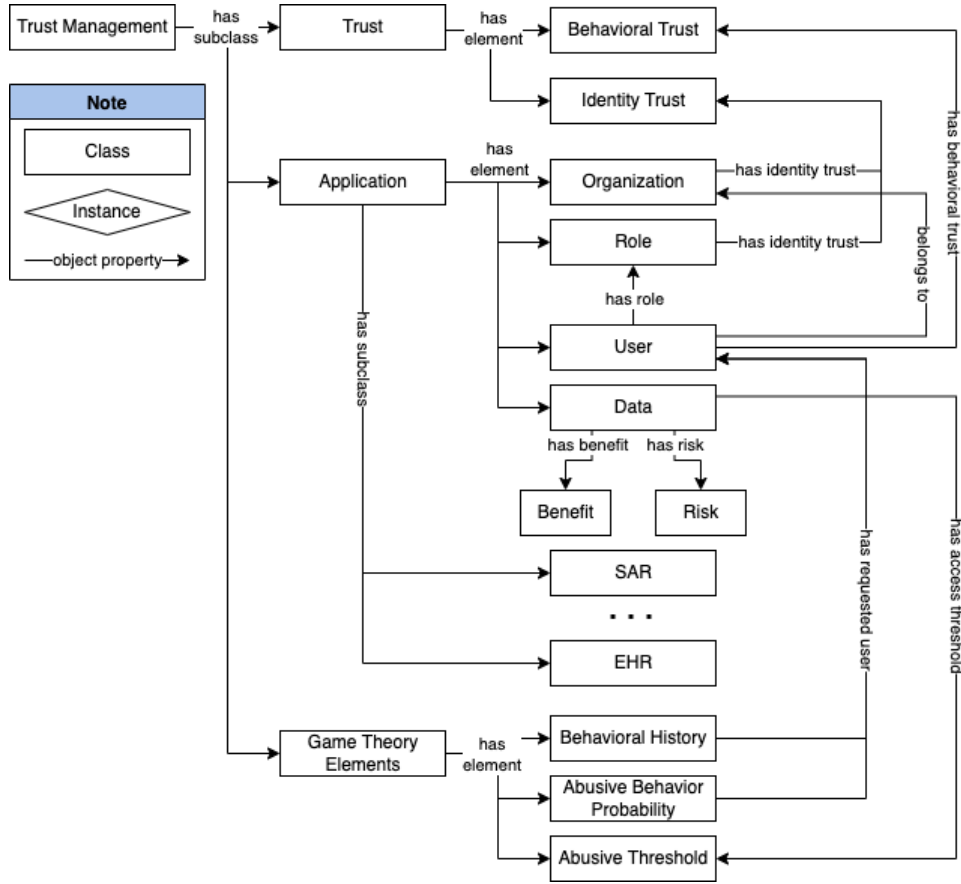


Fig. 1. Trust Management Ontology

other members in a federated system.

### III. METHODOLOGY

In this section, we describe the MATS access control framework in detail. Our framework consists of the following five components:

- MATS Access Control Model (described in III-A) that defines the rules we have included in this framework.
- Trust Management Ontology (shown in Figure 1 and described in III-B) which takes into account a user's and organization's identity-based trust scores and target data's benefit and risks. Users can extend this Trust ontology for their specific scenarios.
- Ontology of the Data Sharing Situation. We have used maritime search and rescue as the situation use case in this paper to validate our approach and have described the maritime SAR Ontology in Section III-C.
- Federated Querying across multiple participants. Section III-D describes our approach to implementing the proposed framework in federated systems.
- Game theory model to dynamically adjust access decisions. Section III-E describes how we use game theory to dynamically adjust the threshold used in access control and update users' trust scores based on users' behavior.

Our framework can be applied to other application domains as well, such as disaster relief or contact tracing, and is part of our ongoing work in this research.

#### A. MATS Access Control Model

We extended the situation-aware access control model in [1] to add consideration of trust, as well as users' behavior, benefits and risks of granting access to specific data. The new model has the following components:

- $U$ : the set of users.
- $R$ : the set of roles.  $R$  has a hierarchical structure.
- $UR \subseteq U \times R$ : the assignment of users to roles.
- $O$ : the set of data that can be shared.
- $PU, PO$ : properties of data or users.
- $SE$ : set of situation expressions typically on properties of users or objects ( $PU, PO$ ).
- $P$ : the set of permissions defined on  $O$ .
- $UT, RT$ : users' or roles' trust scores, which is a combination of multiple aspects, including identity (e.g., credentials of an individual) and behavior (whether the user has normal behavior in history).
- $PM$ : a payoff matrix of each type of data for different players when the other player takes different actions. Since we apply game theory to access control, the two players are data requester and data provider. The possible

actions for provider are whether to grant access or deny access. The possible actions for requester are normal behavior or abusive behavior. More details can be found in Section III-E.

- *UB*: users' behavioral history, including "normal" behavior and "abusive" behavior. A user's behavioral history can often be observed by users in the system or by a system coordinator.
- *TE*: expression on a user's trust score. E.g., a user's combined trust score (as a weighted sum of scores on different aspects of trust)  $\geq$  a user set threshold, which often depends on the confidential level (whether the data item is classified) and veracity (high quality/resolution data may require higher trust threshold) of the requested data item.
- *BE*: expression on a user's behavior. E.g., a user's probability of abusive behavior  $\leq$  a threshold. A user's probability of abusive behavior can be computed from a user's behavioral history (*UB*). The threshold can be computed from the payoff matrix *PM* using game theory.
- $SEUR \subseteq 2^{SE} \times UR$ , the set of situation-aware assignment of roles to users.  $2^{SE}$  is power set of *SE*.
- $DTSERP \subseteq 2^{SE} \times 2^{TE} \times 2^{BE} \times RP$ , the set of permission assignments based on situation, trust, behavior, and role of the user.

*SEUR* and *DTSERP* can be represented as access control rules. Here is a sample access control rule using our model: a user can only have read access to a vessel's location and direction information if the user belongs to the SAR team (role), has a trust score over 1.0, is within 100 KMs from the vessel in distress (situation), and has the probability of abusive behavior below a threshold computed from the payoff matrix.

### B. Trust Management Ontology

Figure 1 illustrates the general trust management ontology. The trust management ontology has three main subclasses - Trust, Application, and Game Theory Elements.

The Trust class has three element classes: Behavioral Trust, Identity Trust, and Trust Threshold. Identity trust represents a user's credentials. Organization and roles can also have identity trust. If a user has no credential, we can also use that user's associated organization or role's identity trust.

Each type of data is associated with a trust threshold which specifies the minimal trust score needed to access the data. One possible way to set the threshold is based on the data item's confidential level, which is consistent with the Bell-LaPadula security model [24] requiring that users at a given security level can only read data at the same or lower security level. For example, we can set the threshold to a high value for classified data and set it to a low value for unclassified data.

The application class has elements representing constituents of a federated data exchange system, including organization, role, user and data. Each data instance has associated benefits and risks. These benefits and risks are used by the game theory model presented in the section III-E to compute the payoff matrix.

The game theory elements class includes three classes: users' behavioral history class, probability of abusive behavior, and a threshold for abusive behavior for each data instance. These three classes are used by a game theory model to dynamically update trust and the system's access control decisions.

A user's behavioral history is used to estimate a user's probability of abusive behavior. For example, a straightforward way to do so is to divide the number of observed abusive behaviors by the total number of observations. A user's behavioral history is also used to update a user's trust score. Using game theory, the system will also compute a threshold for abusive behavior for each data instance from the payoff matrix *PM* given in Section III-E. This threshold will be directly used in access control such that only users whose probability of abusive behavior is less than this threshold will be allowed to access the associated data instance.

### C. SAR Mission Ontology

As a use case example for a federated data-as-a-service application, we modified the ontology for maritime SAR in [3] based on the U.S. Coast Guard SAR manual [25] as well as the general trust ontology developed in Section III-B. Figure 2 shows some major classes in the SAR ontology.

We describe a vessel in distress using following query patterns in a SPARQL query:

```
?vessel rdf:type sar:Vessel .
?vessel sar:hasEmergencyPhase sar:Distress .
```

In the same way, we can query vessels belonging to the Coast Guard Yard in New York as:

```
?vessel rdf:type sar:Vessel .
?vessel sar:hasOrganization ?Org .
?Org sar:isCommandedBy sar:NewYorkCoastGuardYard .
```

### D. Trusted Federated Query Framework

We propose a Situation-Aware Trusted Federated Query Framework to implement our access control model (shown in Figure 3), which extends the framework proposed in [3].

Our proposed framework is efficient yet relatively straightforward. We use query rewriting to allow access control checking to be a part of query evaluation, which can allow users to access many data objects at a time. We also use an existing semantic-web framework without the need to implement additional complex modules such as distributed reasoning. In addition, we varied the architecture (whether trust-related data is stored distributed or centralized) to optimize the system's performance.

We adopt a framework which consists of a mission coordinator (MC) and trusted middleware (TM) at each member of the system. The primary purpose of the Mission Coordinator (MC) is to maintain the data access policy and a master copy of trust scores of users and organizations, as well as users' behavioral data. In our implementation, this copy is stored in an Apache Jena Fuseki SPARQL server [26]. Trusted Middleware at each member can query this master copy when they do not have corresponding information. Trusted Middleware (TM) controls access to data when a user queries

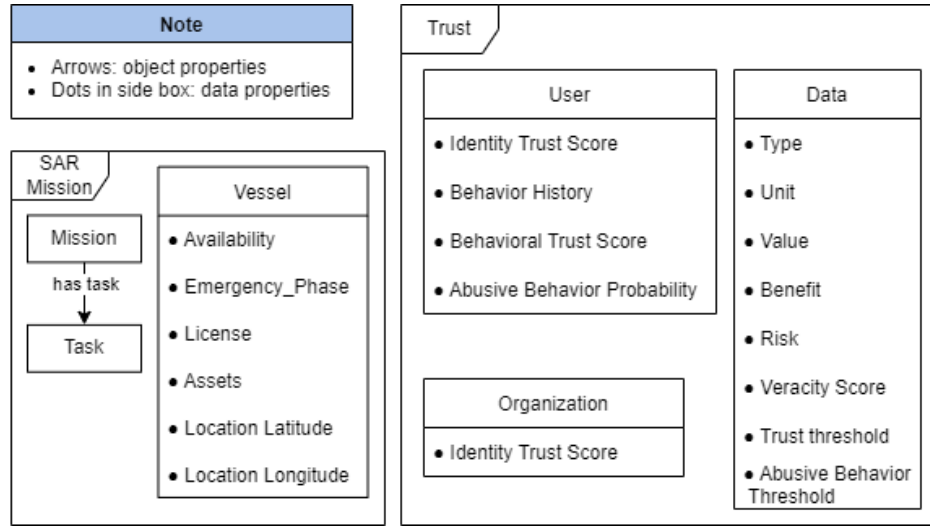


Fig. 2. Major Classes in SAR Ontology

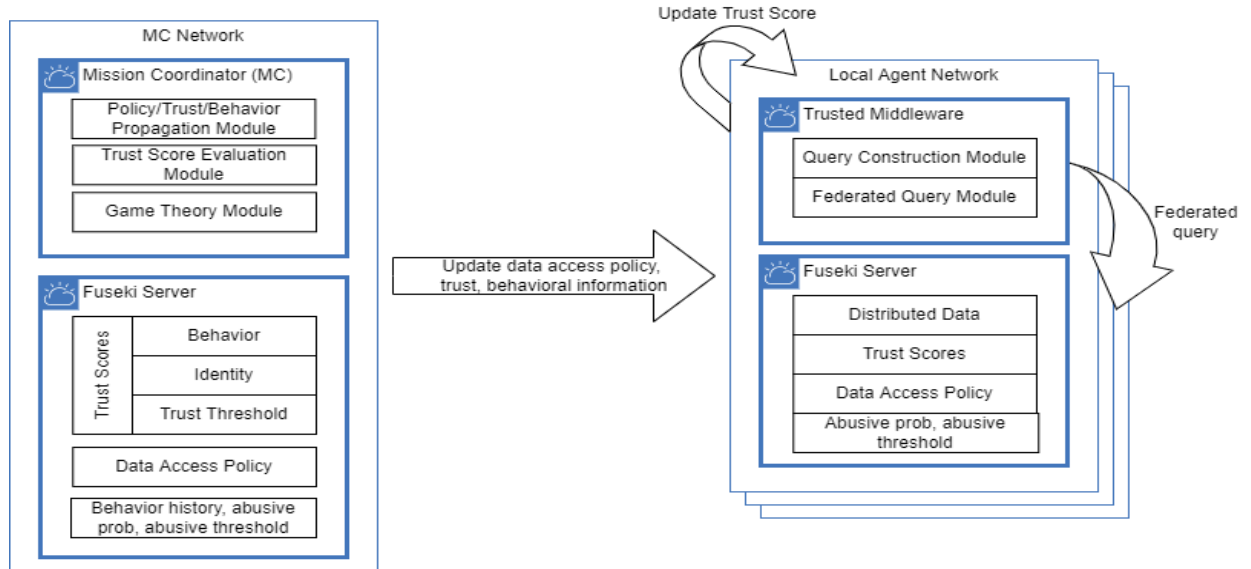


Fig. 3. Trusted Federated Query Framework

data and sends appropriate federated queries to other TMs when the requested data is not available in the local server (Apache Jena Fuseki server in our implementation).

The reason for having this architecture is two-fold: 1) it simplifies management of trust and access control policies by having a centralized mission coordinator; 2) in many applications such as maritime SAR, there is often already a coordinator in the system which can serve as Mission Coordinator. For example, each SAR mission is typically led by a SAR coordinator, who can verify credentials of members, assign trust scores, and gather behavioral data about each member, either by the coordinator itself or reported by members. Next we describe components in MC and TM.

**Mission Coordinator:** Mission Coordinator has a trust evaluation module which assigns identity and behavioral trust

scores to users based on their credentials and behavioral history.

Mission coordinator also uses a game theory model to compute an estimated probability of abusive behavior for each user and a threshold for this probability for all data items, and updates trust scores based on newly observed user behavior. Details can be found in Section III-E. This threshold can be computed offline because it only depends on the payoff matrix, which is static. The probability of abusive behavior, however, needs to be recomputed after each observed user behavior.

The propagation module provides the latest data access policy set by data owners, and propagates changes to access policies, trust scores, or probabilities of abusive behavior and associated threshold to different members.

**Trusted Middleware:** TM consists of two modules: Query

Construction Module and Federated Query Module. TM also has a local Fuseki server which stores local data and a local copy of trust scores, data access policy, probability of abusive behavior and thresholds for users belonging to that member.

**Centralized vs. distributed storage of trust and behavioral data:** trust and behavioral data can be stored centralized at the MC or distributed at each TM.

Centralized storage simplifies updates of such data, but may make the mission coordinator a bottleneck at query execution time because all TMs will need to retrieve trust and behavioral information from the coordinator. On the other hand, a purely distributed solution will not have a bottleneck but will make updating of trust and behavioral data more complex because only the MC is authorized to make such updates (e.g., a member cannot update its own trust scores).

In this paper we use a hybrid solution where MC still maintains a master copy of trust and behavioral data for all users in the system. However, each member's TM stores a local copy of trust scores and derived behavioral information (i.e., probability of abusive behavior and the threshold of this probability computed using game theory) associated with its local users. At querying time, TMs can work together to answer a query without the involvement of MC. When the trust and behavioral data of a specific user changes, the MC will update the master copy first and then push the update to the corresponding MC.

**Query rewriting:** The Query Construction Module rewrites a query in the following steps. First, it adds conditions in data access policies (i.e., sets of situation expressions, trust expressions, and behavioral expressions) set by the data owners to the where clause of the original query. Second, since data to answer each condition may be stored at different members, it looks for the locations of the requested data using ASK queries. Once the locations containing requested data are found, subqueries with SERVICE keywords are added to retrieve data from those locations. The results will then be merged using the UNION operator. Finally it returns requested data in the SELECT clause.

Users can specify access control policies by describing situations - e.g., locations of the data requester - and customized rules consisting of situation, trust, and behavior expressions. For example, a data owner can require the user's trust score to be greater or equal to the trust threshold of each requested data. Also, the owner can dictate that the user's probability of abusive behavior be less or equal to the threshold of abusive behavior computed using game theory.

Source Code 1 is an example constructed query for our SAR use case that returns the current direction of a target vessel if the requester (User\_073) has a probability of abusive behavior less than 0.2, and a trust score of over 2, or a trust score of over 1 if the requester's vessel has a towing license, or the target's vessel is within 110 kms of the vessel in distress. These conditions are checked in a FILTER clause. There are three possible vessels so three subqueries with SERVICE keywords are sent to these vessels and results from these vessels are combined using the union operator.

---

```

SELECT DISTINCT ?t_vessel, ?data, ?value
WHERE {
  sar:User_073 sar:Identity_Trust_Score ?user_idt .
  sar:User_073 sar:Behavioral_Trust_Score ?user_bhv .
  sar:User_073 sar:belongsTo ?org .
  ?org sar:Identity_Trust_Score ?org_idt .
  sar:User_073 sar:Abuse_Prob ?prob .
  sar:User_073 sar:isCrewOf ?u_vessel .
  ?u_vessel sar:License ?license .
  ?u_vessel sar:Location_Latitude ?v_lat .
  ?u_vessel sar:Location_Longitude ?v_long .
  ?t_vessel rdf:type sar:Vessel .

  BIND("37"^^xsd:integer AS ?d_lat)
  BIND("-72"^^xsd:integer AS ?d_long)
  BIND(0.2 * ?user_idt + 0.3 * ?user_idt + 0.5 *
    ↪ ?org_idt AS ?tscore)
  BIND((?d_lat-?v_lat)*(?d_lat-?v_lat) +
    ↪ (?d_long-?v_long)*(?d_long-?v_long) AS
    ↪ ?distance)

  {
    ?t_vessel sar:hasData ?data .
    ?data sar:Type "Current_Direction" .
    ?data sar:Value ?value .
  }
  UNION {
    SERVICE <http://fuseki_1:3030/hmm> {
      ?t_vessel sar:hasData ?data .
      ?data sar:Type "Current_Direction" .
      ?data sar:Value ?value .
    }
  }
  UNION {
    SERVICE <http://fuseki_3:3030/uscg> {
      ?t_vessel sar:hasData ?data .
      ?data sar:Type "Current_Direction" .
      ?data sar:Value ?value .
    }
  }
  UNION {
    SERVICE <http://fuseki_4:3030/usnavy> {
      ?t_vessel sar:hasData ?data .
      ?data sar:Type "Current_Direction" .
      ?data sar:Value ?value .
    }
  }

  FILTER ((?tscore > 2 && ?prob < 0.2) ||
    ↪ (?license="Towing"^^xsd:string && ?tscore > 1 &&
    ↪ ?prob < 0.2) || (?distance <= 110 && ?prob <
    ↪ 0.2))
  FILTER (!isBlank(?data))
}

```

---

Source Code 1. Federated SPARQL query example

**Query Execution:** Since Jena Fuseki servers already support federated query execution, the federated query module just needs to send the rewritten query to the local Jena Fuseki server, which will coordinate with other Jena Fuseki servers specified in the service clauses to execute the query. More specifically, the portion of queries with a SERVICE keyword is sent to the corresponding member for execution. The results are then combined at the local server and returned to users.

Since the rewritten query already contains conditions to enforce access control rules, users can only see results that they are allowed to see.



### E. Game Theory Model

We use game theory to model access control scenarios in a federated data-as-a-service environment, and describe a game model for dynamic access control. We will then present how to compute a threshold for abusive behavior, which will be directly used in access control such that only users with probability of abusive behavior lower than the threshold will be allowed to access a data instance. Finally we will show how to update a user's trust score after each access.

Our game model analyzes the interaction between two players, including the user, who requests data from another party in the federation, and the data provider. Allowing the user to access the data can result in benefits or risks to the data provider, depending on the behavior choice of the user. A user might choose to behave in a normal way and have normal access, which can bring benefits to both players, namely the user and the provider. On the other hand, there might be some temptations for the user to abuse their access privileges to gain extra benefits.

The data provider can maintain the security of the requested data by considering both the benefits and risks of allowing access to the data. For example, sharing the location of a ship in distress can have high benefits to the rescue mission, and at the same time, the risk of sharing such data might be low. However, sharing personal data about crew members of the ship can be highly risky, and may not be necessary for the mission. Therefore, access decisions should consider the benefits and risks of sharing data.

In addition, trust is a key factor for access decisions in a federated environment. Users with high trust can access more confidential data compared to users with low trust. Trust can change according to users' access behavior. Users with normal access behavior are rewarded by increasing their trust scores, resulting in more future access. On the other hand, users may lose their access privileges because of their abuse of access permissions. Users with abusive access behavior are penalized by decreasing their trust scores, which can restrict their future access.

**Game Model for dynamic access control:** For dynamic access decisions in federated data-as-a-service systems, we model the access control scenario using a non-zero-sum co-operative game, in which both players, the user and the data provider, can win if the user chooses normal access [7]. The access control game model can be defined using the following components:

- $u$ : the user who requests data.
- $d$ : the data provider.
- $A_u = \{N, A\}$  is the action set of the user, who can choose normal ( $N$ ) or abuse ( $A$ ) access behavior.
- $A_d = \{G, D\}$  is the action set of the data provider, who can choose to grant ( $G$ ) or deny ( $D$ ) an access request.

Table I shows the payoff matrix for the federated access control game. In each cell the first payoff is for data provider and the second payoff is for user (data requester). Below are notations used in the table:

TABLE I  
PAYOFF MATRIX FOR SAR ACCESS CONTROL

Data Provider	User	
	Normal	Abuse
Grant	$Benefit_d, Reward_u$	$-Risk_d, Benefit_u + ExtraBenefit_u - Penalty_u$
Deny	$-Cost_d, 0$	$0, 0$

- $Benefit_d$ : the benefit that the data provider receives from the user's normal access.
- $Benefit_u$ : the benefit that the user receives from having normal access.
- $Risk_d$ : the risk to the data provider as a result of the user's access abuse.
- $ExtraBenefit_u$ : the extra benefit that the user receives for access abuse.
- $Cost_d$ : the cost to the data provider as a result of denying access to a normal user.
- $Reward_u$ : the reward to the user for choosing normal access.
- $Penalty_u$ : the punishment to the user for choosing access abuse.

In the federated access game, the user's payoff depends on the reward and penalty, which change dynamically with every access permission. Therefore, there is no pure strategy for the game. However, we can obtain the Nash Equilibrium (i.e., no party can increase the expected payoff by using a different strategy) by using mixed strategies (i.e., each party takes an action with a certain probability). Assuming that the data provider grants access with probability  $p$  and denies access with probability  $1 - p$ , then the mixed strategy for the data provider is  $(p, 1 - p)$ . Additionally, if we assume that the user chooses access abuse with probability  $q$ , and normal access with probability  $1 - q$ , then the mixed strategy for the user is  $(q, 1 - q)$ . The total expected utility of the data provider is:

$$U_d = p[q(-Risk_d) + (1 - q)Benefit_d] + (1 - p)[q \times 0 + (1 - q)(-Cost_d)] \quad (1)$$

To maximize data sharing, the data provider can make access decisions using the Nash Equilibrium of the mixed strategies of the user  $(q^*, 1 - q^*)$ .

$$q^* = \frac{(Benefit_d + Cost_d)}{(Risk_d + Benefit_d + Cost_d)} \quad (2)$$

**Computing threshold for abusive behavior:** In practice, the  $q^*$  for Nash Equilibrium may still lead to negative overall payoff for the data provider. So the data provider can use a threshold  $q_t$  that guarantees positive payoff.

$$-q_t \times Risk_d + (1 - q_t)Benefit_d = 0 \quad (3)$$

The threshold  $q_t = \frac{Benefit_d}{Benefit_d + Risk_d}$  can be pre-computed for each data item by the data provider. When a user sends an access request to a data item, we compare the user's probability of abuse and compare it to the threshold of the



requested data item. If the probability of the user's abuse is less than the threshold ( $q \leq q_t$ ), then the access is granted; otherwise, the access is denied. A condition to check whether a user's probability of abusive behavior is less than  $q_t$  will be added in the query rewriting process.

The threshold depends on the benefit and risk of granting access to the data. As the data confidentiality increases, the risk of granting access to the data grows; hence, the access threshold decreases, and the access decision becomes less tolerant.

We estimate a user's probability of abuse  $q$  based on the user's access history. The abuse probability of the user is measured as the number of times the user chooses to abuse access out of the entire access history of the user.

**Trust Update:** The trust of the user is updated after each access based on the user's behavior as well as the risk or benefit utility of the data provider. Assume  $x$  and  $y$  represent the total number of times the user chooses abuse or normal use, respectively. We use the trust update function proposed by [7] to adjust the user behavioral trust  $T(u)$  for the  $x^{th}$  time of abuse and the  $y^{th}$  time of normal use.

$$T(u) \begin{cases} \max(T(u) - \varphi(x)Risk_d, 0) & \text{abuse} \\ T(u) + \emptyset(y)Benefit_d & \text{normal} \end{cases} \quad (4)$$

Where  $\varphi$  and  $\emptyset$  are functions of the times of abuse and normal use.

$$\begin{cases} \varphi(x) = \frac{1}{2}x^2 \\ \emptyset(y) = 2y \end{cases}$$

$\varphi$  is quadratic but  $\emptyset$  is linear because we want to penalize abusive behavior more.

#### IV. RESULTS

##### A. Experiment Setup

**Dataset:** Since there are no publicly available data sets for maritime SAR, we generated synthetic SAR datasets in four different sizes based on the Global Ocean Current Database (GOCD) provided by National Centers for Environmental Information (NCEI), and the National Oceanic and Atmospheric Administration (NOAA) [27]. GOCD integrates ocean current data from various capture methods, resolutions, and formats. We generated four categories of the data - current speed, current directions, east-west component speed, north-south component speed.

Our dataset design includes five different vessels from different organizations. Table III gives the attributes of vessels belonging to various organizations. Situation-aware access control policies, which we cover in the following section, refer to these attributes to describe the precise contexts of data access. Each vessel also records ocean current data such as its direction and speed, which is a time series and useful in SAR missions. Each vessel has 25 users and an evenly distributed amount of ocean current data for every data size, e.g., 250, 2,500, and 25,000. So we generated three data sets with around 1250, 12500, and 125000 triples.

**Queries:** We ran a query where the rewritten query is as given in Source code 1. The original query is asked by a user of a rescue ship to return current direction data of all vessels in the same mission. This query is quite expensive and returns around 70% of data in the database. Also, the shapes of a rewritten query vary in each experimental system structure - centralized, fed-central, or hybrid - because the trust scores and data distributions are different.

**Access Control Rules:** To demonstrate a situation-aware access control policy, we provide three data access policies that cover organizations and users' reputations and behavioral trust of users while considering the context of the data request - in this case, the location of the data requester's vessel. The *FILTER* clause in source code 1 enforces the access control policy of the experiments:

- Rule 1: The weighted sum of trust scores is more significant than two, and the user's abusive behavior probability is less than 0.2 (`?tscore > 2 && ?prob < 0.2`)
- Rule 2: The user's vessel has a towing license, and the weighted sum of the trust score is greater than one, and the user's abusive behavior probability is less than 0.2 (`?license="Towing"^^xsd:string && ?tscore > 1 && ?prob < 0.2`)
- Rule 3: The distance between the user's vessel and the vessel in distress is less than 110, and the user's abusive behavior probability is less than 0.2 (`?distance <= 110 && ?prob < 0.2`)

**Metrics:** We measure the query execution time of running a batch of 100 queries, each from a random user. Each TM created ten threads to run these queries simultaneously. The execution time can be divided into time for query rewriting, which is dominated by using ASK queries to find out vessels that contain relevant data, and the time for executing the rewritten query. We ran each query batch three times and report the average time. We restarted all the TMs between each run to minimize the impact of caching. We did not notice much fluctuation of execution time between the runs.

Since we need to update the trust score and behavioral data about the user who asked the query, we also report the time for trust/behavioral data update. These updates were implemented using SPARQL delete and insert statements (SPARQL does not have update statements).

Assuming that there will be a new observation of a user's behavior for each query the user asks, we also report overall time as the sum of time for query rewriting, execution, and for updating the trust and behavior of the user who asks the query.

**Experiment Design:** We consider three cases.

- 1) Centralized: this is a traditional server-client model for data exchange. In this network, MC has all trust scores and data. This is not a federated system. We use this as a baseline to show the overhead of using a federated system.
- 2) Fed-central: this case has data stored distributed in a federated system but the trust and behavioral data are stored at MC.

TABLE II  
EXPERIMENTAL RESULTS

Data Size (triples)	1,250			12,500			125,000		
System	Centralized	Fed-Central	Hybrid	Centralized	Fed-Central	Hybrid	Centralized	Fed-Central	Hybrid
Query Rewriting(sec)	0	20.93	20.53	0	23.2	22.54	0	27.57	31.87
Query Execution(sec)	1.95	4.9	4.45	9.58	15.47	10.42	124	95.67	105
Trust Update(sec)	7.05	10.77	10.11	26.38	31.9	18.46	236.33	238.67	128
Overall(sec)	9	36.6	35.09	43.716	70.57	51.42	360.33	361.9	265.2
Query rewriting+execution(sec)	1.95	25.83	24.98	9.58	38.67	32.96	124	123.24	136.87

TABLE III  
VESSEL INFORMATION

Organization	Emergency Phase	License	Latitude	Longitude
NOAA	None	None	46.3	-63.4
HMM <sup>1</sup>	None	None	29.5	-65.2
US Navy	None	Towing	31.7	-79.8
US Coast Guard	None	None	43.2	-81.5
MSC <sup>2</sup>	Distress	None	37	-72

- 3) Hybrid: this case has distributed data and hybrid storage of trust scores and behavioral data (i.e., TMs have a local copy of trust scores and behavioral information, and MC has a master copy).

We employed a Docker container for each TM and Apache Jena Fuseki server to simulate five trusted middlewares simultaneously. Each Docker container has a minimum of 3 GB and a maximum of 6 GB RAM capacity. We uploaded data to each Jena Fuseki server with the API provided by Jena Fuseki. One of the TM also serves as the mission coordinator.

### B. System Performance Results

Table II shows the time to execute a batch of 100 queries for the Centralized, Fed-central, and Hybrid cases.

The results show that the Centralized case is the most efficient in terms of query execution, which is expected because data is at one place without the overhead of running queries remotely. The centralized case also does not require ASK queries because all data is at one place so its query rewriting time is negligible so we reported zero.

The gap between Hybrid and Centralized is not drastic though. For the largest data set, if we count both the query rewriting and execution time, Hybrid took 136.87 seconds and Centralized took 124 seconds. The ratio is higher for smaller data sets due to the overhead of query rewriting and federated queries but the absolute differences are within 30 seconds. Note that the time reported is the time to execute 100 concurrent queries, so the overhead of implementing access control and query answering in a federated system is often acceptable.

Interestingly, for the largest data set, Centralized has higher trust update time and overall time than Hybrid. This is

probably due to fact that the centralized case has data from all members, so it is much larger than the data at each member and takes more time to update the trust scores. A federated system like Hybrid allows multiple members to work in parallel.

Between the two federated solutions, Hybrid leads to shorter overall time than Fed-Central for all data sets, and the query execution time of both approaches is similar. Interestingly, we observed that Fed-Central was much slower on trust updates than Hybrid. This is unexpected as the Hybrid solution needs to update both the master copy at the MC and the local copy at TM. We find that this is likely because in Hybrid, queries are distributed at TMs without the involvement of MC but in Fed-Central all query patterns related to trust scores are sent to MC. So MC became a bottleneck in Fed-Central because it needs to query and update many trust scores at the same time, which significantly slows down the updates. This indicates that it is more efficient to keep a local copy of trust and behavioral data at each member to avoid having the MC as a bottleneck.

### C. Game Simulation Analysis

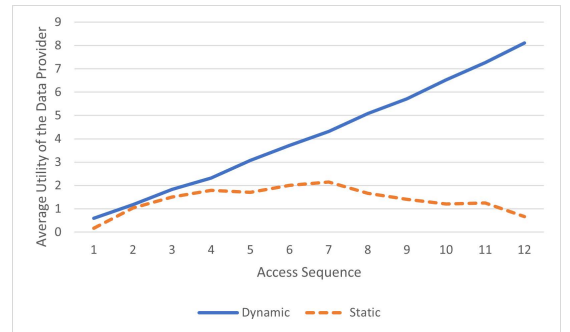


Fig. 4. Comparing Utility between Dynamic and Static Access Control

We have generated data for the SAR scenario to validate the dynamic access control model. We randomly initialize trust data for 1,000 users with 50 consecutive access processes for each user. Access requests are randomly sent to one of four data items that we have defined with different low and high levels of benefit and risk. Assuming users are rational players who try to maximize their payoffs, we can estimate users' choice of abuse or normal use for each access process. The

<sup>1</sup>Hyundai Merchant Marine

<sup>2</sup>MSC Industrial Direct

trust data is updated after each access stage based on the access behavior of users.

We compare our dynamic access control with a static model that uses trust levels to assign access privileges via access rules. Similar to the simulation of the dynamic access, we use random trust scores to generate data for 1,000 users, who can request access to the same data with varying degrees of benefits and risks. However, users' trust cannot be updated in the static access model. To protect the data from users who abuse their access privileges, we apply the grim-trigger strategy, in which we always grant users requests as long as they do not abuse their access permissions. If they do, then all their future access will be denied.

We select a random sample of 100 users with 12 consecutive access stages for each access model. Then we calculate the average utility of the data provider at each access stage and compare the results between the dynamic and the static access model. Figure 4 shows that the average utility of the data provider is higher in the dynamic access model than the static model for all stages of the access process. With dynamic access control, all users can participate in each access stage; however, their access privileges vary depending on their updated trust scores after each access process. If users lose some trust points because they abuse their access privileges, they will not be allowed to access highly confidential data. However, they can still access low-risk data, which can bring more benefits to the data provider. On the other hand, the number of users who can request data declines with the static model at each access stage. In this model, the access control strategy prevents users who abuse their access privileges from accessing any more data regardless of the benefits to the data provider or the rescue mission.

In addition, the results indicate that the utility of the data provider consistently increases with each access stage in the dynamic access control compared to the static access model. The access threshold in the dynamic access control ensures the Nash Equilibrium for each access decision based on the benefit and risk of the requested data. However, the access control strategy in the static model grants all access permissions for users with trust levels that satisfy the access rules without considering the benefits and risks they could bring to the data provider.

## V. CONCLUSION

This paper proposes a multi-aspect and adaptive trust-based situation-aware access control framework for federated Data-as-a-Service systems. This framework is applicable to applications such as disaster relief, maritime SAR, or contact tracing. The main contributions are threefold: 1) we propose a trust model that considers multiple aspects including users and organizations' identity as well as users' behavior; 2) we use a game model to set thresholds in the access control model and dynamically adjust trust scores based on users' behavior; 3) we show that the proposed framework can be implemented efficiently using a semantic-web based architecture. Experimental results show that this solution is efficient and that the

dynamic game model leads to a higher payoff for the data provider than a static model.

As future work, we will apply our solution to other use cases such as contact tracing. It will be also interesting to compare query rewriting based solutions with solutions using distributed reasoning.

## ACKNOWLEDGMENT

This research was partially supported by a DoD supplement to the NSF award 1747724, Phase I IUCRC UMBC: Center for Accelerated Real time Analytics (CARTA), and Office of Naval Research grant # N00014-18-1-2452 and N00014-18-1-2453.

## REFERENCES

- [1] S. S. Yau, Y. Yao, and V. Banga, "Situation-aware access control for service-oriented autonomous decentralized systems," in *Autonomous Decentralized Systems*. IEEE, 2005, pp. 17–24.
- [2] D. Beimel and M. Peleg, "Using owl and swrl to represent and reason with situation-based access control policies," *Data & Knowledge Engineering*, vol. 70, no. 6, pp. 596–615, 2011.
- [3] S. Oni, Z. Chen, A. Crainiceanu, K. P. Joshi, and D. Needham, "A framework for situation-aware access control in federated data-as-a-service systems based on query rewriting," in *2020 IEEE International Conference on Services Computing (SCC)*. IEEE, 2020, pp. 1–11.
- [4] Department of Justice, "Department of justice reminds the public to be aware of fraud when disaster strikes and report it to the national center for disaster fraud," 2019. [Online]. Available: <https://www.justice.gov/opa/pr/departement-justice-reminds-public-be-aware-fraud-when-disaster-strikes-and-report-it-nation-0>
- [5] G. D. Kutz, "Hurricanes katrina and rita disaster relief : improper and potentially fraudulent individual assistance payments estimated to be between \$600 million and \$1.4 billion," 2006.
- [6] R. Bhatti, E. Bertino, and A. Ghafoor, "A trust-based context-aware access control model for web-services," *Distributed and Parallel Databases*, vol. 18, no. 1, pp. 83–105, 2005.
- [7] N. Helil, A. Halik, and K. Rahman, "Non-zero-sum cooperative access control game model with user trust and permission risk," *Applied Mathematics and Computation*, vol. 307, pp. 299–310, 2017.
- [8] Y. Weihong, "Research on maritime search and rescue decision-making ontology model," in *2009 International Conference on Environmental Science and Information Application Technology*, vol. 2. IEEE, 2009, pp. 140–142.
- [9] W. Yu and R. Li, "Maritime search and rescue ontology construction based on protege," in *2009 International Conference on Information Engineering and Computer Science*. IEEE, 2009, pp. 1–3.
- [10] D.-y. Kim, L. Elluri, and K. P. Joshi, "Trusted compliance enforcement framework for sharing health big data," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 4715–4724.
- [11] D.-y. Kim and K. P. Joshi, "A semantically rich knowledge graph to automate hipaa regulations for cloud health it services," in *2021 7th IEEE Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE, 2021, pp. 7–12.
- [12] R. Walid, K. P. Joshi, S. G. Choi, and D.-y. Kim, "Cloud-based encrypted ehr system with semantically rich access control and searchable encryption," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 4075–4082.
- [13] L. Sun, H. Wang, J. Yong, and G. Wu, "Semantic access control for cloud computing based on e-healthcare," in *Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2012, pp. 512–518.
- [14] A. Kayes, J. Han, and A. Colman, "An ontological framework for situation-aware access control of software services," *Information Systems*, vol. 53, pp. 253–277, 2015.
- [15] S. Oulmakhzoune, N. Cuppens-Boulahia, F. Cuppens, and S. Morucci, "Privacy policy preferences enforced by sparql query rewriting," in *2012 Seventh International Conference on Availability, Reliability and Security*, 2012, pp. 335–342.

- [16] A. Padia, T. Finin, A. Joshi *et al.*, "Attribute-based fine grained access control for triple stores," in *3rd Society, Privacy and the Semantic Web-Policy and Technology workshop, 14th International Semantic Web Conference*, 2015.
- [17] S. Chakraborty and I. Ray, "Trustbac: integrating trust relationships into the rbac model for access control in open systems," in *Proceedings of the eleventh ACM symposium on Access control models and technologies*, 2006, pp. 49–58.
- [18] F. Feng, C. Lin, D. Peng, and J. Li, "A trust and context based access control model for distributed systems," in *2008 10th IEEE International Conference on High Performance Computing and Communications*. IEEE, 2008, pp. 629–634.
- [19] J. Bernal Bernabe, J. L. Hernandez Ramos, and A. F. Skarmeta Gomez, "Taciote: multidimensional trust-aware access control system for the internet of things," *Soft Computing*, vol. 20, no. 5, pp. 1763–1779, 2016.
- [20] J. He, S. Ma, and B. Zhao, "Analysis of trust-based access control using game theory," *International Journal of Multimedia & Ubiquitous Engineering*, vol. 8, no. 4, pp. 15–24, 2013.
- [21] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu, "A Survey of Game Theory as Applied to Network Security," in *2010 43rd Hawaii International Conference on System Sciences*. IEEE, 2010.
- [22] H. Hu, G. J. Ahn, Z. Zhao, and D. Yang, "Game theoretic analysis of MultiParty Access Control in online social networks," in *Proceedings of ACM Symposium on Access Control Models and Technologies, SACMAT*. Association for Computing Machinery, 2014, pp. 93–102.
- [23] Y. Wang, L. Tian, and Z. Chen, "Game analysis of access control based on user behavior trust," *Information*, vol. 10, no. 4, 2019.
- [24] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," MITRE CORP BEDFORD MA, Tech. Rep., 1973.
- [25] U.S. Department of Homeland Security United States Coast Guard, "U.S. COAST GUARD ADDENDUM TO THE UNITED STATES NATIONAL SEARCH AND RESCUE SUPPLEMENT (NSS) To The International Aeronautical and Maritime Search and Rescue Manual (IAMSAR)."
- [26] Apache Jena, "Apache jena: A free and open source java framework for building semantic web and linked data applications," accessed May 30, 2022. [Online]. Available: <https://jena.apache.org/>
- [27] Sun, Charles; US DOC/NOAA/NESDIS > National Centers for Environmental Information (2018). NCEI Standard Product: Global Ocean Currents Database (GOCD) (NCEI Accession 0171666). Version 3.0. NOAA National Centers for Environmental Information. Dataset. Accessed Jan 26, 2022.