

A Policy Framework for Data Fusion and Derived Data Control

Jerry den Hartog
Eindhoven University of Technology
j.d.hartog@tue.nl

Nicola Zannone
Eindhoven University of Technology
n.zannone@tue.nl

ABSTRACT

Recent years have seen an exponential growth of the collection and processing of data from heterogeneous sources for a variety of purposes. Several methods and techniques have been proposed to transform and fuse data into “useful” information. However, the security aspects concerning the fusion of sensitive data are often overlooked. This paper investigates the problem of data fusion and derived data control. In particular, we identify the requirements for regulating the fusion process and eliciting restrictions on the access and usage of derived data. Based on these requirements, we propose an attribute-based policy framework to control the fusion of data from different information sources and under the control of different authorities. The framework comprises two types of policies: *access control policies*, which define the authorizations governing the resources used in the fusion process, and *fusion policies*, which define constraints on allowed fusion processes. We also discuss how such policies can be obtained for derived data.

CCS Concepts

•Security and privacy → Formal security models; Access control; Authorization; •Information systems → Mediators and data integration;

Keywords

Access control; usage control; data fusion

1. INTRODUCTION

Data fusion has been gaining attention in recent years as an important aspect of big data and situation-awareness. Situation-awareness is based on the idea that a system should be flexible and adaptable to different circumstances [27]. The current context and system status (the ‘situation’) determines the desired actions in complex application domains (e.g., critical systems) and knowledge of this situation is thus a fundamental part of any decision making

process. To create this knowledge, huge amounts of data are often collected from a variety of sources. The challenge being tackled in data fusion is how to extract knowledge by “interpreting” and combine data coming from heterogeneous sources with different conceptual and contextual knowledge representations (e.g., body sensor networks, motion sensors, satellites, social media).

Although several models and methods have been proposed to address the problem of data fusion (see [3] for a survey), the security aspects of the fusion process are often overlooked by existing approaches. In this respect, we have identified two main concerns: (i) the control of the data fusion process and (ii) the protection of derived data and the control of its usage.

The data used in a fusion process can be acquired by sources controlled by different authorities. Each authority might impose constraints on the access and usage of their data. For instance, an authority may require that its classified information is not fused with information at a lower integrity level or specify with which (type of) information its data can be fused and for which purpose.

Moreover, new data elements can be created as the result of a fusion process. These data elements might reveal sensitive information concerning the data used for their creation [2, 14, 30]. The amount of information that can be inferred from derived data can either increase, decrease or remain unchanged depending on the fusion process [21]. Thus, constraints on the access and usage of derived data should be defined by considering the constraints on both the data elements used for their creation and the (type of) fusion process that has been applied to obtain them.

Several frameworks and mechanisms have been proposed for the protection of sensitive information in the literature [8, 15, 20, 29]. In particular, attribute-based policy languages provide a fine-grained solution to regulate the access and usage of sensitive information which is essential for situation awareness. However, existing solutions often assume that each piece of information is under the control of a single principal and do not account that pieces of information can be transformed and combined with other information to create “new” information. Moreover, existing solutions do not consider restrictions on which data can be fused together and on which fusion process can be applied to certain data. Therefore, they do not provide a comprehensive approach to regulate a fusion process and protect derived data.

In this work we investigate the problem of data fusion and derived data control. First, we identify the basic requirements for regulating the fusion process and for

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ABAC'16, March 11, 2016, New Orleans, LA, USA.

© 2016 ACM. ISBN 978-1-4503-4079-3/16/03...\$15.00

DOI: <http://dx.doi.org/10.1145/2875491.2875492>

defining restrictions on the access and usage of derived data. Based on these requirements, we propose an attribute-based policy framework to control the fusion process in which data from different information sources and under the control of different authorities have to be fused. The framework comprises two types of policies: *access control policies*, which define the authorizations governing the resources used in the fusion process, and *fusion policies*, which define constraints on allowed fusion processes, i.e. on how and with what data can be fused. In this paper, we focus on the specification of such policies. Note that a typical fusion process will involve multiple policies of both types and from different authorities. We show how access control and fusion policies can be set for derived data based on the constraints regulating the access and usage of the resources (data elements and fusion process) used for their creation. We demonstrate the applicability of the framework using a running example within the military domain, which is based on a case study from the IN4STARS 2.0 project.

The remainder of the paper is organized as follows. The next section introduces a case study, which is used as a running example through the paper. Section 3 identifies requirements for data fusion and derived data control. Section 4 discusses related work. Section 5 presents a formalization of data fusion. Section 6 presents how access control and fusion policies can be specified, and Section 7 discusses how they can be set for derived data. Section 8 shows an application of the policy framework to the case study. Finally, Section 9 concludes the paper providing directions for future work.

2. MOTIVATING EXAMPLE

This section introduces a case study in the military domain that has been studied within the IN4STARS 2.0 project and is used as a running example throughout the paper. All names and facts introduced in the scenario are purely fictional.

The IN4STARS case study is set in Petraceros, a poor, unstable country controlled by an authoritarian military dictatorship. In recent years, complaints regarding the dictatorial regime governing Petraceros have grown. The economy is greatly damaged by the crisis and, while people are starving, the wealth and power of the leaders grow constantly. Recently, more and more people begin promoting the idea of a government change through social networks, catching the attention of the masses. The government's attempts to impose censorship on communication channels have resulted in a reaction from the people, who have started protesting against the regime. The demonstrations and protests have intensified and spread to the entire country. As a response, the government have mobilized the army. Even though the protests are not violent, the army has received the green light to shoot civilians. The tension has escalated, and the army has become increasingly violent against the growing number of protesters. In response to these events, the NATO has created a Combined Joint Task Force (CJTF), named CJTF-ALPHA, to enforce peace in the territory by suppressing the violent army. The CJTF is a multinational operation composed of combat and intelligence units belonging to different NATO and Partnership for Peace (PfP) members, including the Netherlands, Estonia and Sweden.

Our motivating scenario focuses on a "hearts and minds" campaign within the CJTF, which aims to gain trust from

the citizens. A Dutch unit of the CJTF is appointed to gather intelligence in a village, increase the trust in the coalition troops by talking to the locals and hearing if things have changed after the combat actions of the previous days. Before approaching the village, the team leader of the unit contacts a tactical intelligence officer of the Swedish army to get intelligence about possible riots in the village.

The tactical intelligence officer uses an intelligence analysis tool to assess the threat of riots in the mission area using information from different sources. (See also Figure 1 in Section 5.) Riots are often characterized by crowds of people in a certain area. To this end, the tactical intelligence officer requests data collected by motion sensors deployed in the area by the Estonian army. The tactical intelligence officer also requires an imagery analysis report, indicating the presence of riots based on the analysis of images. His request is forwarded to an imagery analyst of the Command and Control Center of the Dutch army located in The Hague. In turn, the imagery analyst collects images from the specified area. Images are retrieved from a UAV deployed by the Estonian army in the area and from MAJIIC, a database of surveillance and reconnaissance resource under the control of the Netherlands. The imagery analyst examines the received images and creates a report about the presence of riots in the specified area, which is sent to the tactical intelligence officer. Finally, the tactical intelligence officer requests a sentiment report from an OSINT analyst of the Swedish army. The OSINT analyst uses various web-crawling and visual analytic tools to estimate the sentiment expressed toward entities of interest. In particular, he runs crawlers on open source blogs and tweets associated with the inhabitants of the mission area. The OSINT analyst provides the tactical intelligence officer the sentiment report. The tactical intelligence officer processes the collected information and fuses them in a riot report, which is sent to the team leader.

The scenario shows that international cooperation and orchestration is needed to ensure the success of the mission. In particular, information from heterogeneous sources should be collected and fused to enable situation awareness and thus properly act in a certain circumstance. Information sources, however, can be under the control of different authorities. Due to the high sensitivity of data, each authority wants to regulate the access and usage of its data. This becomes more challenging when data is fused with other data as it can be difficult to assess, especially automatically, how much information can be inferred from the derived data. In the next section we identify the requirements needed to control a data fusion process and to protect the usage of derived data.

3. REQUIREMENTS FOR DATA FUSION AND DERIVED DATA CONTROL

Data fusion requires taking into account various security aspects [25]. In this section we identify and discuss the requirements for data fusion and derived data control. Before discussing these requirements, we introduce a few terms that are used throughout the paper.

Data controller: the entity responsible for the security of the data under its control. This entity defines who can access a data element and how it can be fused.

ID	Requirement
R1	Data processors should be authorized to perform the fusion process.
R2	Data processors should be authorized to access input data.
R3	Data can only be used in certain fusion processes.
R4	Data can only be fused with certain other data.
R5	Authorizations for derived data should account for the authorizations for the data used for their creation.
R6	Authorizations for derived data should account for the fusion process used for their creation.
R7	Restrictions on the fusion of derived data should account for the restriction on the data used for their creation.
R8	Restrictions on the fusion of derived data should account for the fusion process used for their creation.

Table 1: Requirements for data fusion and derived data control

Data processor: the entity who processes and fuses a set of data elements.

Data consumer: the entity who accesses a data element.

There are several requirements that should be satisfied for the protection of sensitive information in collaborative dynamic and distributed systems [9, 27]. In this work, we are mainly interested in requirements for data fusion and control of the derived data. We have identified two main groups of requirements (Table 1). The first group focuses on requirements regulating the fusion process and the pre-existing resources (**R1** to **R4**); the second group focuses on the use, including refusion, of derived data (**R5** to **R8**).

To fuse data, a data processor is required to be authorized to perform the fusion process (**R1**) and have access to the input data elements (**R2**). In our scenario tools for image analysis and for the creation of imagery analysis reports are made available to the CJTF-ALPHA by the Dutch Command and Control Center in The Hague. The use of these tools, however, is restricted to imagery analysts of the Dutch Army. This is captured by requirement **R1**. On the other hand, requirement **R2** states that only users with access to the input data elements can execute the fusion process. For instance, in our scenario, the Estonian Control and Command Center (the data controller for UAV images and motion data) imposes that only imagery analysts assigned to the CJTF-ALPHA can access images gathered by UAVs deployed in the Petraceros area. Note that the access to the input data elements is defined by the corresponding data controller.

The requirements so far constrain who may perform the fusion. It can also be that a data element should only be fused in certain ways, giving additional types of constraints. In particular, data controllers may impose constraints on how their data can be fused (**R3**), i.e. which fusion process can be applied to their data. In our scenario, the Estonian Control and Command Center allows motion data to be fused into a riot report but not into a vehicle count/movement report. Moreover, the fusion of certain data elements with specific types of data may be undesirable (**R4**). For instance, the Swedish army can impose that riot reports generated using its intelligence analysis tool cannot be fused with civilian data.

The requirements discussed above impose constraints on who can perform a given fusion process and on the data elements that can be used in that process. Similar constraints need to be put on the derived data newly

created as output of the fusion process. The second group of requirements provides guidelines on how these constraints should be inferred for derived data elements. We argue that any policy framework for data fusion needs to provide inference of policies for derived data addressing these requirements. Derived data can reveal sensitive information concerning the data used in the fusion process. Thus, data controllers can impose restrictions on who can access data elements derived from their data and how such data elements can be refused. Requirements **R5** and **R6** focus on the specification of authorizations for derived data.

Derived data elements can be subsequently fused with other data elements. For instance, in our scenario, the imagery analysis report is fused with motion data and a sentiment report in order to create a riot report. As such we also need requirements **R7** and **R8**, which focus on restrictions concerning the fusion of derived data. Together these four requirements state that constraints on the access and usage of derived data should depend on the constraints associated with input data, data fusion process, or both.

For instance, consider the riot report in our running example. The report can be associated with a constraint stating that its disclosure to a non-NATO country or a non-PIP country is forbidden. These constraints can be general and apply to all riot reports as they are a sensitive type of information. In other words, the access is dictated by the type of derived data and/or fusion process used to create it. Another possibility is that the given riot report is confidential due, for instance, to the mission area, whereas riot reports about other areas are not. In this case, the access restriction is dictated by the input data. We will come back to this distinction in Section 7.

In the next section we review how the identified requirements have been addressed in the literature. Then, in Section 6 and 7, we give a policy framework enabling the specification of constraints implementing these requirements.

4. RELATED WORK

Controlling the usage of digital resources has been attracting increasing attention over the years. In this work, we are particularly interested in the security implications of data fusion. To this end, we review existing literature with respect to the requirements identified in Section 3. A summary of this analysis is presented in Table 2.

The protection of sensitive information is typically achieved using access control. Access control relies on policies defining which actions a subject is allowed to perform on a given

	Data fusion				Derived data				Legend
	R1	R2	R3	R4	R5	R6	R7	R8	
Access Control	✓	✓	✗	✗	✗	✗	✗	✗	✓ satisfied
Data Fusion	✗	✗	✓	✓	✗	✗	✗	✗	✓ partially satisfied
Secure multi-party computation	✗	✓	✓	✗	✗	✗	✗	✗	✗ not satisfied
Information Flow	✗	✓	✗	✗	✓	✗	✗	✗	
Data declassification	✗	✓	✗	✗	✓	✓	✗	✗	
Access Control + Information Flow	✓	✓	✗	✗	✓	✗	✗	✗	
Atluri and Gal [2]	✓	✓	✗	✗	✓	✗	✗	✗	
Zannone et al. [30]	✓	✓	✗	✗	✓	✓	✗	✗	
Scalavino et al. [21]	✓	✓	✓	✗	✓	✓	✗	✗	

Table 2: Comparison of existing approaches with respect to the types of constraints identified in Section 3.

object. However, existing access control solutions [8, 15, 20] require policies to be specified in advance and do not provide support to automatically capture authorizations for derived data. Moreover, they are not able to specify and enforce constraints on which data can be fused together. A few data fusion approaches [10, 13] use policies to control which information can be fused together. However, these proposals mainly focus on quality aspects and do not consider security requirements for the fusion process. Secure multi-party computation [29] can be seen as a solution to the problem of data fusion, in which parties can jointly compute a function over their inputs, while keeping these inputs private. However, this solution neither assesses the amount of information concerning the original data that is revealed as result of the fusion process nor determines constraints on the usage of derived data.

The problem of controlling which information is disclosed by derived data is often addressed within information flow [18]. In particular, information flow aims to determine whether the output of a function leaks sensitive information to unauthorized entities. However, information flow control systems are very rigid as they prevent any flow of information to a lower security level, regardless of the fusion process. To address this issue, some researches [14, 17, 21, 24] have studied the problem of data declassification. Some approaches assume declassification is performed directly by data owners [14] or at programming language level [17]. However, these approaches only check whether the information flow complies with the given policies and do not address the problem of protecting the information created as result of the usage. Moreover, they only account for the security classification of data and do not support reasoning on other types of attributes (e.g., location).

Some proposals complement access control with some form of flow control [6, 12, 16, 19, 22]. For instance, some approaches [12, 22] associate an access control list with each object and propagate it together with the information in the object. Other approaches [6, 16, 19] mainly protect against unauthorized outflow of information. However, these proposals are not expressive enough to model and infer complex fusion constraints.

To the best of our knowledge, only a few studies give attention to the control and protection of information resulting from data fusion. Atluri and Gal [2] introduce the notion of derived authorizations, i.e. authorizations for derived data. They assume that derived data are

obtained through reversible transformations of some input data and compute derived authorizations based on the union of the authorizations for the original data. Zannone et al. [30] also account for the fusion process to determine the authorization for derived objects. In particular, they distinguish two types of transformation functions: functions for which derived data reveal the information contained in the original data and functions for which derived data provide less information than the one obtained by accessing the original data. However, as in [2] this work provides a simplistic derivation for authorizations, which is not able to capture the complexity and subtlety of real fusion processes. Scalavino et al. [21] propose a labeling system to control derived data. This labeling system determines the security label of derived data based on the security labels of original data and transformation function used to create them. Security labels include a security domain to ensure that derived data are correctly protected on the basis of all resources that contributed to their creation. However, similarly to other proposals based on information flow, the work in [21] is limited to authorization constraints based on the security classification of data and is not able to deal with arbitrary constraints.

As shown in Table 2 only few proposals are able to regulate how data can be fused (**R3** and **R4**). Moreover, the state of affair concerning the control of the usage of derived data is rather limited. Some methods provide a way to control the access to derived data based on the authorizations for the input data (**R5**) and transformation function (**R6**). However, no solutions have been proposed to derive constraints on how derived data can be reused (**R7** and **R8**). In this work we aim at a policy framework that is able to regulate the fusion process as well as to protect derived data and regulate their usage. Our policy framework is general in that it allows the specification and reasoning on arbitrary security constraints that encompass the specific needs of an application domain. In this respect, solutions as the one proposed in [21] can be integrated in our framework to derive specific types of constraints.

5. DATA FUSION

Data fusion aims to combine data from multiple sources to achieve improved accuracy or increase the confidence in the information. In particular, data fusion allows more specific inferences than those that could be achieved by the

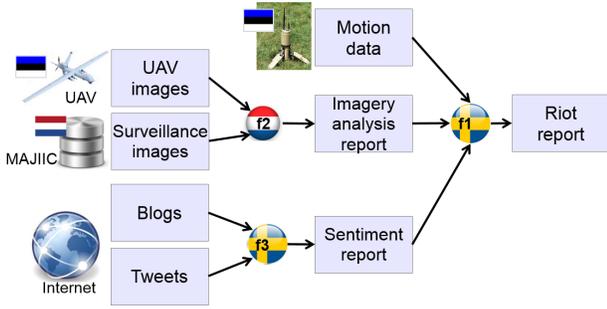


Figure 1: Data fusion model

use of a single data source alone [7]. Several models and techniques for data fusion have been proposed (see [3] for a survey). For the purpose of this work, we abstract from the specific data fusion techniques and represent the process of fusing data through data fusion functions. Intuitively, a data fusion function denotes the transformation of a (set of) data element(s) into a new data element.

DEFINITION 1 (DATA FUSION FUNCTION). Let \mathcal{D} be the set of data elements. A data fusion function (with arity n) is a function $f : \mathcal{D}^n \rightarrow \mathcal{D}$. A derived data element d is the result of applying a data fusion function f to data elements $d_1, \dots, d_n \in \mathcal{D}$, i.e. $d = f(d_1, \dots, d_n)$. We use \mathcal{F} to denote the set of data fusion functions.

A data fusion function can be graphically represented through a model that specifies which data elements are aggregated and the outcome of the fusion process. It is worth noting that derived data elements can subsequently be fused with other data elements.

EXAMPLE 1. Figure 1 presents a graphical representation of the fusion processes underlying the scenario of Section 2. The riot report is created through data fusion function f_1 by fusing motion data, an imagery analysis report and a sentiment report. In turn, the imagery analysis report is created through data fusion function f_2 by fusing UAV images acquired by the Estonian army and surveillance images retrieved from MAJIIC; and the sentiment report is created through data fusion function f_3 by fusing blogs and tweets from the Internet. These fusion functions are under the control of different authorities: f_1 and f_3 are provided by Swedish army, and f_2 by the Dutch army.

As the example above shows, the simple fusion function model may describe complex human driven processes that can be combined into even more complex processes. For example, function f_2 captures imagery analysts using their experience to manually examine images collected from the UAV and MAJIIC databases, merging images using image editing software, highlighting key items visible on an image, etc. However, from the perspective of the mission, this is all seen as one service. The data fusion model and the data control system match this level of abstraction by treating this as one function.

Our definition of data fusion function places no restrictions on how input data elements are used in the fusion process. Input data may thus capture parameters that influence the process but do not directly become part of the

fused data, such as the features used in sentiment analysis for emotion classification and opinion extraction. Note that such parameters can also be encoded by using different functions. For instance, two fusion processes using the same sentiment analysis tools but with different sets of input features can be modeled as two distinct fusion functions.

Similarly to [21], we assume that there exists a predefined set of data fusion functions and these functions are known by all parties within the coalition. This assumption is twofold. First, data controllers should know what functions are available, thus enabling them to select which are allowed to be used on their data. Second, data controllers should know how these functions transform the data so they can specify adequate protection measures for derived data.

6. DATA FUSION CONTROL

To implement the requirements regulating the data fusion process, we employ two types of policies: *access control* policies, which are used to determine whether a data processor can access the data to be fused, and *fusion* policies, which are used to determine how data can be fused. In addition, data fusion functions are also associated with an access control policy, which determines whether a data processor is allowed to execute that function.

In this work we adopt an attribute-based policy language for the specification of access control and fusion policies. This choice is motivated by the fact that attribute-based access control has been shown to be suitable to handle authorizations in open and distributed systems [9, 23]. The basic idea underlying attribute-based access control is that subjects, objects, actions and the system environment are represented by attributes. In particular, attributes specify properties of subjects, objects, actions and environment that can be used for authorization decision making.

DEFINITION 2 (ATTRIBUTES). Let S be the set of subjects, O the set of objects, A the set of actions, and E denote the environment. Every subject $s \in S$, object $o \in O$, action $a \in A$ has a set of attributes. In addition, E is described by a set of attributes. Let denote $Attr_S$ the set of subject attributes, $Attr_O$ the set of object attributes, $Attr_A$ the set of action attributes and $Attr_E$ the set of environment attributes. The set of all attributes is denoted by $Attr$, i.e. $Attr = Attr_S \cup Attr_O \cup Attr_A \cup Attr_E$. Each attribute $attr \in Attr$ is associated with a data type τ which determines the set of values that can be assigned to the attribute (called attribute domain and denoted by $dom(\tau)$) and with a set of binary predicate operators that can be applied to values of this type (denoted by $preds(\tau)$).

Attribute-based access control policies use attributes to describe access rules. For the sake of compactness and clarity, hereafter, we introduce a simple abstract notation for the specification of attribute-based access control policies, which suffices for the scope of this work. Note that policies expressed in our notation can be easily mapped to existing attribute-based access control languages like XACML [15].

DEFINITION 3 (ACCESS CONTROL POLICY). An access control policy P is an expression of the form:

$$\begin{aligned} P &= \text{Permit} \mid \text{Deny} \mid (T, P) \mid (ca, [P, \dots, P]) \\ T &= attr \phi v \mid T \wedge T \mid T \vee T \mid \neg T \end{aligned}$$

Permit and Deny represent the policies that permit every request and deny every request respectively; (T, P) is a targeted policy where T is a target; $(ca, [P, \dots, P])$ represents a composite policy where ca is a policy combining algorithm. A target T is constructed from primitive constraints using \wedge (AND), \vee (OR), and \neg (NOT). Primitive constraints are of the form $attr \phi v$, where $attr \in Attr$, $v \in \text{dom}(\tau)$ with τ the data type of $attr$ and $\phi \in \text{preds}(\tau)$. We use \mathcal{P} to denote the set of access control policies.

Intuitively, the target of an access control policy defines its applicability space, i.e. the set of access requests to which the policy applies. The notation introduced in the definition above can be used for the specification of access control policies for both data elements and data fusion functions. In particular, in access control policies for data elements, data fusion functions are treated as actions; on the other hand, in access control policies for data fusion functions, data fusion functions are treated as objects and the action refers to the right to execute a certain function.

Several policy combining algorithms have been proposed in the literature [8, 11, 15]. Intuitively, these algorithms define procedures to evaluate composite policies based on the order of the policy elements and priorities between access decisions. In this work, we do not impose constraints on which combining algorithms should be used. For instance, one can use the combining algorithms provided by XACML v3 [15]: permit-overrides, deny-overrides, deny-unless-permit, permit-unless-deny, first-applicable and only-one-applicable. We refer to [15] for a definition of such algorithms.

EXAMPLE 2. In the scenario of Section 2, the Estonian Command and Control Center wants to restrict the access to images gathered by Estonian UAVs deployed in the field. In particular, access rights that data consumers can have depend on the location where the images were taken. Specifically, the Estonian Command and Control Center restricts the access to images gathered by UAVs deployed in the Petraceros area to imagery analysts assigned to CJTF-ALPHA operations, while the access to UAV images gathered in Europe is restricted to imagery analysts of NATO and PfP countries. These authorization requirements can be represented by the following access control policy:

```
(permit-overrides, [
  (type = "image"  $\wedge$  source = "UAV"  $\wedge$ 
   role = "imagery analyst",
   (permit-overrides, [
     (mission = "CJTF-ALPHA"  $\wedge$  area = "Petraceros",
      Permit),
     ((country  $\in$  NATO  $\vee$  country  $\in$  PfP)  $\wedge$  area = "Europe",
      Permit)])),
  Deny])
```

with $role, country, mission \in Attr_S$, $type, source, area \in Attr_O$, $action-id \in Attr_A$. NATO and PfP represent the sets of NATO and PfP member states respectively. (Note that technically $country \in NATO$ is a shorthand for $\forall x \in NATO \text{ country} = x$.)

In attribute-based access control a request to use a resource is expressed as a collection of attributes. Here we focus purely on the specification of the policies and assume it is clear how requests can be evaluated against policies.

Although attribute-based access control allows the specification of fine-grained access control policies, it is subject to attribute hiding attacks [28], where a user deliberately hides some attributes to gain additional rights. This problem, however, is orthogonal to the scope of this work. Thus, in this work we assume that all needed attributes are available for policy evaluation.

To control the data fusion process, we also introduce fusion policies. Intuitively, these policies impose constraints on the data elements that can be fused with a given data element and on the fusion functions that can be used over a given data element.

DEFINITION 4 (FUSION POLICY). Let \mathcal{D} be the set of data elements and \mathcal{F} the set of data fusion functions. A fusion policy Q has the form $(T, \{(D_1, F_1), \dots, (D_m, F_m)\})$ where T is a target, $D_i \subseteq \mathcal{D}$ is a set of data elements and $F_i \subseteq \mathcal{F}$ is a set of data fusion functions. A target T is constructed from primitive constraints using \wedge (AND), \vee (OR), and \neg (NOT). Primitive constraints are of the form $attr \phi v$, where $attr \in Attr_O$, $v \in \text{dom}(\tau)$ with τ the data type of $attr$, and $\phi \in \text{preds}(\tau)$. We use \mathcal{Q} to denote the set of fusion policies.

Intuitively, a fusion policy $(T, \{(D_1, F_1), (D_2, F_2)\})$ states that the data elements specified in the target T can only be fused with a set of data elements D_1 using a data fusion function in F_1 or with a set of data elements D_2 using a data fusion function in F_2 . It is worth noting that, differently from the target of access control policies, the target of fusion policies only constraints the data elements to which a policy is applicable. This is because here the focus is on how a data element can be fused rather than on the authorizations governing the fusion process.

EXAMPLE 3. The Dutch Command and Control Center requires that images from MAJIIC can only be fused with data elements that have a classification of secret or higher. However, it allows the fusion of images retrieved from MAJIIC with UAV images at any secrecy level using its tools for image analysis (e.g., f_2 in Example 1 and, indirectly, using f_1). A fusion policy encoding these requirements can be defined as follows:

```
(type = "image"  $\wedge$  source = "MAJIIC",
  {{{d  $\in$   $\mathcal{D}$  | d.classification  $\geq$  secret}},  $\mathcal{F}$ },
  {{{d  $\in$   $\mathcal{D}$  | d.source = "UAV"}}, { $f_1, f_2$ }})
```

where notation $d.classification$ denotes the secrecy level of data element d and $d.source$ the source of data element d . Note that the Dutch Command and Control Center does not impose any constraint on the data fusion function when images are fused with other secret information; thus, in this case we use \mathcal{F} to indicate that all fusion functions can be used in the fusion process concerning MAJIIC images.

We stress that, given a data element, only the access control and fusion policies defined by the data controller should be used to determine whether the data element can be used in a fusion process. Similarly, fusion functions are under the control of a certain authority. Thus, only the access control policy defined by the authority controlling the use of a fusion function should be evaluated to determine whether the function can be used by the data processor.

7. DERIVED DATA CONTROL

When a new data element is created as the result of a fusion process, it can reveal sensitive information about the data elements that have been fused and the fusion function (and its input parameters) used to create it; thus, its access and usage should be regulated. To this end, we associate an access control policy and a fusion policy to derived data elements. Ideally, such policies should be automatically created from the policies associated with the data elements which have been fused and with the data fusion function used in the fusion process. Unfortunately, in practice these constraints typically need to be defined on a case-by-case basis.

Thus, instead of pursuing a fully automated approach for the generation of policies for derived data, we propose to associate two policy templates to data fusion functions: an *access control template* and *fusion template*. These templates are then instantiated for a given derived data element based on the actual data elements that have been fused as well as on the parameters given as input to the fusion function.

An access control template is essentially an access control policy containing elements that can be customized during the fusion process. It defines constraints that depend on the fusion process (represented by the fusion function) and on the type of derived data obtained from its execution. This is necessary to capture situations in which derived data reveal more information than the data elements used to create them. For instance, derived data can provide more accurate and reliable information by reducing the level of uncertainty of the data [4]. In addition, access control templates can contain two types of customizable elements: policy references and attribute mapping functions. Policy references are used to refer to the policies of the data elements that are fused. Attribute mapping functions are used to define customizable primitive constraints in the target of an access control template. Intuitively, the constraints forming the target of the template might be instantiated based on the input data elements and fusion function.

DEFINITION 5 (ATTRIBUTE MAPPING FUNCTION). *Let $Attr$ be the set of attributes. An (n -ary) attribute mapping for an attribute $attr \in Attr$ with data type τ is a function $\gamma : dom(\tau)^n \rightarrow dom(\tau)$. We use Γ to denote the set of attribute mapping functions.*

The attributes of derived data created with some fusion function will typically depend on the attributes of the input data being fused and of the fusion function itself. Thus, we use attribute mapping functions for the specification of the target of policy templates associated to the fusion function. These attribute mapping functions are similar to fusion functions except that they fuse ‘meta-data’, i.e. the attributes.

EXAMPLE 4. *Consider an imagery analysis report d obtained by analyzing a collection of surveillance images D_1 and UAV images D_2 using a data fusion function f . Each image is associated with a classification label from a lattice with a dominance relation among the labels. Depending on the type of report (i.e., the fusion function used to create the report) and, consequently, on the information contained in the report, a different security classification can be assigned to*

the report. We distinguish whether the original images are contained in the report and whether a sensitive conclusion is drawn in the report:

1. *The imagery analysis report contains the original images but does not contain a sensitive conclusion. As such we want it to be accessible to those who may access all of the images. Then, the classification of the report can be defined as:*

$$d.classification = \bigsqcup_{d_i \in D_1 \cup D_2} d_i.classification$$

where symbol \bigsqcup represents the least upper bound.

2. *The imagery analysis report does not contain the original images and does not contain a sensitive conclusion. As such we want it to be accessible to all who can access any of the images. Then, the classification of the report can be defined as:*

$$d.classification = \bigsqcap_{d_i \in D_1 \cup D_2} d_i.classification$$

where symbol \bigsqcap represents the greatest lower bound.

3. *The imagery analysis report contains the original images and draws a sensitive conclusion. Then, the classification of the report can be defined as:*

$$d.classification = \left(\bigsqcup_{d_i \in D_1 \cup D_2} d_i.classification \right) \sqcap f.classification$$

4. *The imagery analysis report does not contain the original images but does draw a sensitive conclusion. Then, the classification of the report can be defined as:*

$$d.classification = \left(\bigsqcap_{d_i \in D_1 \cup D_2} d_i.classification \right) \sqcap f.classification$$

We now have the machinery to define access control templates.

DEFINITION 6 (ACCESS CONTROL TEMPLATE). *Let \mathcal{D} be the set of data elements, \mathcal{F} the set of data fusion functions and Γ the set of attribute mapping functions. The access control template Λ associated to an n -ary data fusion function $f \in \mathcal{F}$ has the form:*

$$\begin{aligned} \Lambda &= \text{Permit} \mid \text{Deny} \mid R_i \mid (T, \Lambda) \mid (ca, [\Lambda, \dots, \Lambda]) \\ T &= attr' \phi \gamma(d_1.attr, \dots, d_n.attr, f.attr) \mid T \wedge T \mid \\ &T \vee T \mid \neg T \end{aligned}$$

Permit and Deny represent the access control template that permit every request and deny every request respectively; R_i is the reference to the access control policy associated to the i th argument of f ; (T, Λ) is a targeted template where T is a target; $(ca, [\Lambda, \dots, \Lambda])$ represents a composite template where ca is a combining algorithm. A target T is constructed from primitive constraints using \wedge (AND), \vee (OR), and \neg (NOT). Primitive constraints are of the form $attr' \phi \gamma(d_1.attr, \dots, d_n.attr, f.attr)$, where $d_1, \dots, d_n \in \mathcal{D}$

are the arguments of f , $attr', attr \in Attr$ with the same attribute domain τ , $\phi \in \text{preds}(\tau)$, $\gamma \in \Gamma$ and notation $x.attr$ is used to represent the value of $attr$ for $x \in \mathcal{D} \cup \mathcal{F}$.

An instantiated template should apply to the derived data element and only to this element. To this end, we instantiate an access control template as a targeted policy where the target is used to restrict the applicability of the template to the derived data element. By setting this element as the sole target of the instantiated policy we ensure it only applies to this element. It is worth noting that the target used in policies being referenced may contain constraints specific to the input data elements which may not be satisfied by the derived element. To ensure these policies do apply to the derived data element we remove these constraints when instantiating a policy reference. Finally, we also evaluate the attribute mapping functions when instantiating the policy.

DEFINITION 7 (ACCESS CONTROL TEMPLATE INSTANTIATION). Let Λ be the access control template associated to an n -ary data fusion function $f \in \mathcal{F}$ and $d = f(d_1, \dots, d_n)$ a derived data element, $d_1, \dots, d_n \in \mathcal{D}$, with $P_1, \dots, P_n \in \mathcal{P}$ the access control policies associated to d_1, \dots, d_n respectively. To instantiate template Λ with respect to d we first instantiate targets in the template by replacing every primitive constraint $attr' \phi \gamma(d_1.attr, \dots, d_n.attr, f.attr)$ in the template by $attr' \phi v$ where $v = \gamma(d_1.attr, \dots, d_n.attr, f.attr)$; we use $\llbracket \Lambda \rrbracket$ to denote the result. Next, we instantiate the policies P_i ($i = 1, \dots, n$); we create ‘context-free’ policies $P_i[d_i]$ by replacing every primitive constraint in the policy P_i as follows:

$$(attr \phi v)[d_i] = \begin{cases} (attr \phi v) & \text{if } attr \notin Attr_O, \text{ else} \\ True & \text{if } attr \phi v \text{ holds for } d_i \\ False & \text{otherwise} \end{cases}$$

We then replace all policy references R_i by the context-free policies $P_i[d_i]$. Finally, we set the result to only apply to d resulting in instantiated policy:

$$(object-id = d.object-id, \llbracket \Lambda \rrbracket [R_1/P_1[d_1], \dots, R_n/P_n[d_n]])$$

EXAMPLE 5. In our running example, the Dutch Command and Control Center provides a fusion function f_2 to create imagery analysis reports based on surveillance images and UAV images. This function can be executed by an imagery analyst to determine whether there is a riot in a given area. The Dutch Command and Control Center wants to restrict the access to imagery analysis reports created using f_2 to NATO/PfP imagery analysts and tactical intelligence officers who have a clearance high enough to access the analyzed images in addition to the constraints imposed by the policy associated to UAV images. This authorization requirement can be represented by the following access control template:

(deny-overrides, [
 ((role = “tactical intelligence officer” \vee
 role = “imagery analyst”) \wedge
 (country \in NATO \vee country \in PfP) \wedge
 clearance $\geq \prod_{d_i \in UAV \cup MAJIC} d_i.classification, Permit)$,
 RUAV])

If we instantiate this template using the policy for UAV images in Example 2, we obtain

(object-id = “ReportI22”, (deny-overrides, [
 ((role = “tactical intelligence officer” \vee
 role = “imagery analyst”) \wedge
 (country \in NATO \vee country \in PfP) \wedge
 clearance \geq “secret”, Permit),
 (permit-overrides, [
 (role = “imagery analyst”,
 (permit-overrides, [
 (mission = “CJTF-ALPHA”, Permit),
 (country \in NATO \vee country \in PfP, Permit)]))],
 Deny)]))

where ReportI22 is the id of the derived imagery report and the highest classification of images is “secret”.

As shown in Example 1, derived data can be subsequently fused with other data elements. To control the usage of derived data, we use fusion templates. Intuitively, a fusion template specifies constraints concerning the refusion of derived data.

DEFINITION 8 (FUSION TEMPLATE). Let \mathcal{D} be the set of data elements and \mathcal{F} the set of data fusion functions. A fusion template Φ for an n -ary fusion function $f \in \mathcal{F}$ has the form

$$\Phi = (D, F) \mid R_i \mid \Phi \cup \Phi \mid \Phi \cap \Phi$$

where $D \subseteq \mathcal{D}$ and $F \subseteq \mathcal{F}$ and R_i denotes a references to the constraints associated to the data elements used as the i -th argument of f .

The target of a fusion policy only considers the data element to which it applies. For an instantiation of a fusion template this is always exactly the fused data object. As such no target needs to be considered in the template.

As for access control templates, fusion templates have to be instantiated with respect to the data elements used in the fusion process.

DEFINITION 9 (FUSION TEMPLATE INSTANTIATION). Let Φ be the fusion template for an n -ary fusion function $f \in \mathcal{F}$ and $R_1, \dots, R_n \in \mathcal{C}$ policy references. Given a derived data element $d = f(d_1, \dots, d_n)$ such that $d_1, \dots, d_n \in \mathcal{D}$ and $Q_i = (T_i, C_i) \in \mathcal{Q}$ the fusion policy associated to $d_i \in \{d_1, \dots, d_n\}$, the instantiation of Φ with respect to d_1, \dots, d_n , is

$$(object-id = d.object-id, \Phi[(R_1, \dots, R_n)/(C_1, \dots, C_n)])$$

where $\Phi[\bar{R}/\bar{C}]$ is given by:

$$\begin{aligned} R_j[(R_1, \dots, R_n)/(C_1, \dots, C_n)] &= C_j \\ (D, F)[\bar{R}/\bar{C}] &= \{(D, F)\} \\ (C_1 \cup C_2)[\bar{R}/\bar{C}] &= C_1[\bar{R}/\bar{C}] \cup C_2[\bar{R}/\bar{C}] \\ (C_1 \cap C_2)[\bar{R}/\bar{C}] &= \{(D_1 \cap D_2, F_1 \cap F_2) \mid (D_1, F_1) \in \\ &\quad C_1[\bar{R}/\bar{C}], (D_2, F_2) \in C_2[\bar{R}/\bar{C}]\} \end{aligned}$$

Note that we assume that the access control and fusion templates for a given fusion function as well as the attribute mapping functions used in the access control template are defined by the authority controlling the function and are fixed; i.e. the same operations on the data but with a different template are considered to be a different function. In fact, in open and dynamic scenarios like ours, it is unrealistic to assume that all authorities will reach a

Data	Controller	Access control policy	Fusion policy
Motion	Estonian CCC	$(type = \text{"motion"} \wedge area = \text{"Petraceros"} \wedge role = \text{"tactical intelligence officer"} \wedge mission = \text{"CJTF-ALPHA"}, \text{Permit})$	$(type = \text{"motion"} \wedge area = \text{"Petraceros"}, \{\{d \in \mathcal{D} \mid d.area = \text{"Petraceros"}\}, \{f_1\}\})$
UAV	Estonian CCC	See Example 2.	$(type = \text{"image"} \wedge source = \text{"UAV"} \wedge area = \text{"Petraceros"}, \{\{d \in \mathcal{D} \mid d.area = \text{"Petraceros"}\}, \{f_1, f_2\}\})$
MAJIIC	Dutch CCC	$(type = \text{"image"} \wedge source = \text{"MAJIIC"}, (\text{deny-overrides}, [(clearance \geq secret, \text{Permit}), \text{Deny}]))$	See Example 3.
Blogs	Blog provider	$(type = \text{"blog"}, \text{Permit})$	$(type = \text{"blog"}, \{\mathcal{D}, \mathcal{F}\})$
Tweets	Twitter	$(type = \text{"tweet"}, \text{Permit})$	$(type = \text{"tweet"}, \{\mathcal{D}, \mathcal{F}\})$

Table 3: Data elements and associated policies

Function	Controller	Access control policy	Access control template	Fusion template
f_1	Swedish CCC	$(object-id = \text{"f}_1"} \wedge role = \text{"tactical intelligence officer"} \wedge mission = \text{"CJTF-ALPHA"} \wedge action-id = \text{"execute"}, \text{Permit})$	$(\text{deny-overrides}, [R_1, (clearance \geq d_2.classification \sqcap d_3.classification \wedge mission-area \supseteq \cap(d_1.area, d_2.area), \text{Permit})])$	$(\{d \in \mathcal{D} \mid d.source \neq \text{"civilian"}\}, \mathcal{F})$
f_2	Dutch CCC	$(object-id = \text{"f}_2"} \wedge role = \text{"imagery analyst"} \wedge country = \text{"NL"} \wedge action-id = \text{"execute"}, \text{Permit})$	See Example 5.	$R_1 \cap R_2 \cap (\mathcal{D}, \{f_1\})$
f_3	Swedish CCC	$(object-id = \text{"f}_3"} \wedge role = \text{"OSINT analyst"} \wedge mission = \text{"CJTF-ALPHA"} \wedge action-id = \text{"execute"}, \text{Permit})$	$(mission = \text{"CJTF-ALPHA"}, \text{Permit})$	$(\mathcal{D}, \mathcal{F})$

Table 4: Fusion functions (see also Figure 1) and associated policies

complete and precise semantic alignment before becoming operative [26]. Moreover, we assume that templates are known by potential data controllers. This allows data controllers to select which fusion functions are acceptable for the processing of their data by using the fusion policy (see Definition 4) to only allow functions with acceptable templates.

8. DEMONSTRATION

In this section we combine all components and apply them to the case study introduced in Section 2. Recall that the team leader of a Dutch unit, participating in the CJTF-ALPHA, requests intelligence about possible riots in a village to which they are assigned in order to gather intelligence and talk to the locals. To provide this information a tactical intelligence officer of the Swedish army fuses an imagery report and a sentiment report, which themselves are derived data elements, with motion data.

We have two types of resources: data elements and fusion functions. Associated with each data element we have a fusion policy in addition to an access control policy. For the original data elements (i.e., motion data, UAV images, MAJIIC surveillance images, blogs and tweets posted on the Internet), these policies are defined by the corresponding data controller as shown in Table 3. With the access control policies we achieve requirement **R2**, whereas the fusion policies cover requirements **R3** and **R4**.

The different reports are created using fusion functions. For a fusion function, in addition to an access control policy, we have the information needed to create access control and fusion policies for derived data, namely an access control template and a fusion template (see Table 4). Note

that the access control policy achieves requirement **R1**, the access control template achieves **R5** and **R6** and the fusion template achieves **R7** and **R8**.

We can observe that the imagery analyst of the Command and Control Center (CCC) of the Dutch army is allowed to use fusion function f_2 (see Table 4) and also to access the needed UAV and surveillance images (see Table 3). Similarly, the OSINT analyst can use f_3 and can access its required input data elements. In this example function f_3 is a specific instance of the sentiment report creation resource dedicated to this mission, reflected in its access control template. The policies placed on the reports created using those fusion functions are listed in Table 5. From this table, together with Table 4, we can observe that the tactical intelligence officer can create the riot report (using f_1). The access control template of this report introduces geographic constraints on the position of the requester (represented by subject attribute *mission-area*). These constraints can be encoded, for instance, in GeoXACML [1]. This standard enables the specification of areas and operations on these areas within XACML. Here we adopt a simplified notation: \supseteq is used to denote an area is contain in another, \cap to denote the attribute mapping function intersecting areas and ‘Area-P1’ to denote the result of intersecting the motion sensor and UAV flight areas in this specific case. According to the riot report policy (see Table 5), the Dutch unit leader can obtain this report.

9. CONCLUSION

In this paper we have presented an attribute-based policy framework for data fusion and derived data control. The framework uses access control policies to define the autho-

Report	Derived access control policy	Derived fusion policy
Sentiment	$(object-id = \text{"Reports42"}, (mission = \text{"CJTF-ALPHA"}, Permit))$	$(object-id = \text{"Reports42"}, \{(D, F)\})$
Imagery analysis	See Example 5.	$(object-id = \text{"ReportI22"}, \{(d \in D \mid d.area = \text{"Petraceros"}, \{f_1\})\})$
Riot	$(object-id = \text{"ReportR5"}, (deny-overrides, [(role = \text{"tactical intelligence officer"} \wedge mission = \text{"CJTF-ALPHA"}, Permit), (clearance \geq secret \wedge mission-area \supseteq \text{"Area-P1"}, Permit)]))$	$(object-id = \text{"ReportR5"}, (\{d \in D \mid d.source \neq \text{"civilian"}\}, F))$

Table 5: Derived data elements and associated policies

rizations governing the fusion process, and fusion policies to define constraints on how data can be fused. We have shown how these policies can be defined for derived data based on the data elements and fusion function used for their creation. In particular, we have introduced the notion of policy template. Policy templates contain customizable elements which are instantiated with respect to the data elements and fusion function used in the fusion process.

The work presented in this paper opens several directions for future work. The proposed framework offers a general approach to define the constraints in the target of access control policies for derived data from the constraints in the target of the policies associated to the input data. Each type of constraints (e.g., classification, geolocation) may require different attribute mapping functions for the inference of “combined” constraints. The proposed framework can be complemented with a library of attribute mapping functions, thus facilitating the definition of policy templates.

The framework allows the automated instantiation of policy templates with respect to the data elements used in the fusion process. This, however, may result in an authority not being fully conscious about the effects of combining their policies with the policies defined by other authorities. To this end, our policy framework can be complemented with frameworks that help understand the impact of instantiated policy template on the access and usage of data elements derived from their data. These frameworks can be used, for instance, to analyze and compare policies at design time [28] or to provide feedback to an authority when the decision enforced by the system differs from the constraints it has explicitly specified for its data [5].

In this work we have studied how to protect derived data by inferring policies for derived data based on the input data and fusion function. In some cases, military organizations need to create information having (at most) a certain secrecy level as such information has to be disclose to possibly untrusted parties or parties with a low clearance. Thus, an interesting direction for future work is to investigate methods for selecting fusion functions which provide fused data at the desired secrecy level.

10. ACKNOWLEDGMENTS

This work has been partially funded by the EDA project IN4STARS2.0, the ITEA3 projects FedSS and M2MGrid, the EU FP7 project AU2EU, and the Dutch national program COMMIT under the THeCS project.

11. REFERENCES

- [1] OGC Geospatial eXensible Access Control Markup Language (GeoXACML) 3.0 Core. OGC Discussion Paper, Open Geospatial Consortium, 2013.
- [2] V. Atluri and A. Gal. An authorization model for temporal and derived data: Securing information portals. *ACM Trans. Inf. Syst. Secur.*, 5(1):62–94, 2002.
- [3] J. Bleiholder and F. Naumann. Data fusion. *ACM Comput. Surv.*, 41(1):1:1–1:41, 2009.
- [4] J. J. Clark and A. L. Yuille. *Data Fusion for Sensory Information Processing Systems*. Kluwer Academic Publishers, 1990.
- [5] S. Damen, J. den Hartog, and N. Zannone. CollAC: Collaborative access control. In *Proceedings of International Conference on Collaboration Technologies and Systems*, pages 142–149. IEEE, 2014.
- [6] E. Ferrari, P. Samarati, E. Bertino, and S. Jajodia. Providing flexibility in information flow control for object oriented systems. In *Proceedings of Symposium on Security and Privacy*, pages 130–140. IEEE, 1997.
- [7] D. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.
- [8] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Trans. Database Syst.*, 26(2):214–260, 2001.
- [9] S. P. Kaluvuri, A. I. Egner, J. den Hartog, and N. Zannone. SAFAX - An Extensible Authorization Service for Cloud Environments. *Frontiers in ICT*, 2(9), 2015.
- [10] D. Langlois and E. Croft. A low-level control policy for data fusion. In *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 37–42, 2001.
- [11] N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin. Access control policy combining: Theory meets practice. In *Proceedings of ACM Symposium on Access Control Models and Technologies*, pages 135–144. ACM, 2009.
- [12] C. McCollum, J. Messing, and L. Notargiacomo. Beyond the pale of MAC and DAC-defining new forms of access control. In *Proceedings of Symposium on Research in Security and Privacy*, pages 190–200. IEEE, 1990.
- [13] J. Michelfeit and T. Knap. Linked Data Fusion in ODCleanStore. In *Proceedings of the ISWC Posters & Demonstrations Track*, CEUR Workshop Proceedings 914. CEUR-WS.org, 2012.
- [14] A. C. Myers and B. Liskov. Protecting privacy using the decentralized label model. *ACM Trans. Softw. Eng. Methodol.*, 9(4):410–442, 2000.

- [15] OASIS XACML Technical Committee. eXtensible Access Control Markup Language (XACML) Version 3.0. Oasis standard, OASIS, 2013.
- [16] F. Paci and N. Zannone. Preventing information inference in access control. In *Proceedings of ACM Symposium on Access Control Models and Technologies*, pages 87–97. ACM, 2015.
- [17] B. P. S. Rocha, S. Bandhakavi, J. den Hartog, W. H. Winsborough, and S. Etalle. Towards static flow-based declassification for legacy and untrusted programs. In *Proceedings of Symposium on Security and Privacy*, pages 93–108. IEEE, 2010.
- [18] A. Sabelfeld and A. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications*, 21(1):5–19, 2003.
- [19] P. Samarati, E. Bertino, A. Ciampichetti, and S. Jajodia. Information flow control in object-oriented systems. *IEEE Transactions on Knowledge and Data Engineering*, 9(4):524–538, 1997.
- [20] P. Samarati and S. De Capitani di Vimercati. Access control: Policies, models, and mechanisms. In *FOSAD*, LNCS 2171, pages 137–196. Springer, 2001.
- [21] E. Scalavino, V. Gowadia, and E. Lupu. A labelling system for derived data control. In *Data and Applications Security and Privacy XXIV*, LNCS 6166, pages 65–80. Springer, 2010.
- [22] A. Stoughton. Access flow: A protection model which integrates access control and information flow. In *Proceedings of Symposium on Security and Privacy*, pages 9–18. IEEE, 1981.
- [23] H. Takabi, J. Joshi, and G.-J. Ahn. Security and privacy challenges in cloud computing environments. *IEEE Security Privacy*, 8(6):24–31, 2010.
- [24] J. Thomas, F. Cuppens, and N. Cuppens-Boulahia. Consistency policies for dynamic information systems with declassification flows. In *Information Systems Security*, LNCS 7093, pages 87–101. Springer, 2011.
- [25] B. Thuraisingham. Secure sensor information management and mining. *IEEE Signal Processing Magazine*, 21(3):14–19, 2004.
- [26] D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle. Reputation-based ontology alignment for autonomy and interoperability in distributed access control. In *Proceedings of IEEE International Conference on Computational Science and Engineering*, pages 252–258. IEEE, 2009.
- [27] D. Trivellato, N. Zannone, M. Glaundrup, J. Skowronek, and S. Etalle. A semantic security framework for systems of systems. *Int. J. Cooperative Inf. Syst.*, 22(1), 2013.
- [28] F. Turkmen, J. den Hartog, S. Ranise, and N. Zannone. Analysis of XACML policies with SMT. In *Principles of Security and Trust*, LNCS 9036, pages 115–134. Springer, 2015.
- [29] A. C. Yao. Protocols for secure computations. In *Proceedings of Annual Symposium on Foundations of Computer Science*, pages 160–164. IEEE, 1982.
- [30] N. Zannone, S. Jajodia, F. Massacci, and D. Wijesekera. Maintaining privacy on derived objects. In *Proceedings of Workshop on Privacy in the Electronic Society*, pages 10–19. ACM, 2005.