A Policy System for Control of Data Fusion Processes and Derived Data

Elisa Bertino
Dept of Computer Science
Purdue University
West Lafayette, IN, USA
bertino@purdue.edu

Dinesh Verma
Distributed AI Dept
IBM TJ Watson Research Center
Yorktown Heights, NY, USA
dverma@us.ibm.com

Seraphin Calo
Distributed AI Dept
IBM TJ Watson Research Center
Yorktown Heights, NY, USA
scalo@us.ibm.com

Abstract— **The paper proposes an attribute-based policy framework for a coalition setting in which multiple parties provide data to be used in data fusion processes while at the same time retaining control of how their own data are used in these processes. The framework consists of three main types of policies: (a) access control policies – these allow one to specify controls on the fusion process (e.g., which user can use which data fusion tool) and on the input data to the fusion process; (b) fusion policies – these allow one to specify whether data needs to be pre-processed before being used (for example, whether data must be anonymized before being used, or encrypted and thus fusions must be performed on encrypted data); and, (c) derived data usage policies – these allow one to specify who is authorized to access the data resulting from the fusion. As all these policies are attribute-based policies, they support high-level, flexible, and expressive policy specifications. The paper also briefly discusses technologies for supporting policy enforcement and novel approaches supporting the automatic generation of policies.**

Keywords—Attribute-based Access Control, Data Privacy, Data Security, Generative Policies

I. INTRODUCTION

Data fusion techniques are used in a large variety of applications, such as web entity discovery [1], identity linkage across social networks [2], sensor data assessment [3], data integration [4], and multimodal biometrics authentication [5]. A major issue in many such applications is represented by privacy, as data used for data fusion may be privacy sensitive. Indeed, very early research [6] has shown that even when personally identifiable information is removed from a dataset, by correlating different datasets one can re-identify the removed records. Data confidentiality is also critical. Many organizations are interested in sharing the knowledge obtained by pooling together their datasets; however, they may not always be able to share in the clear their datasets with other organizations. Therefore, techniques have been developed for privacy-preserving record linkage [7]. Finally, data integrity is critical to ensure that data used by fusion processes has not been tampered with and is high quality [8].

In addition to those requirements the problem of security and privacy for data fusion processes is further complicated by recent edge computing trends [9] by which fusions can be collaboratively performed at devices, rather than at centralized locations. Because of the heterogeneity of devices and the variety of security capabilities they may support, identifying where to perform which steps within a fusion process is critical.

To date the problem of security for data fusion processes and the control on derived data has not been widely addressed. It is clear that providing comprehensive solutions requires to combine a lot of different techniques and approaches.

In this paper we aim at addressing (in part) such a gap by focusing on a policy-based management approach to drive the secure and privacy-preserving executions of data fusion processes. Policy-based management uses declarative high-level policies to drive the behavior of the managed parties and processes. It has been widely applied to access control [10] and network management [11]. However, its application to security and privacy of data fusion is relatively unexplored. The goal of the paper is thus to identify the types of policy, referred to as *policy domains*, that are relevant for a comprehensive policy-based management approach for data fusion, provide definitions of such policies, and identify enforcement approaches for these policies.

The paper is organized as follows. Section II introduces a motivating example. Section III introduces an abstract notion of fusion processes and an attribute-based model characterizing the entities involved in fusion processes. Section IV categorizes the different policy domains needed in order to address the requirements for security and privacy. Section V mentions various mechanisms that can be applied to enforce such policies. Section VI discusses novel approaches for policy generation and its applications to data fusion policies. Finally, Section VII discusses future work.

II. MOTIVATING EXAMPLE

Assume that several organizations (military, civilian, law enforcement agencies, and NGO) share data to monitor

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

refugee transportation (both legal and illegal) across the Mediterranean Sea – each with the goal of improving their own operations. Relevant applications vary a lot, ranging from identifying potential terrorists trying to infiltrate into the EU by disguising themselves as refugees, identifying EU citizens who went to fight abroad and are returning to the EU, to medical preparedness and food provisioning for rescue ships. It is clear that whereas all those applications may benefit from sharing and fusing data, some data may be sensitive either because of privacy or because they are obtained from highly sensitive sources (that thus need to be protected). We now introduce three fictitious scenarios that highlight some of the requirements for secure and privacy-preserving data fusion.

Scenario 1. *Passenger ship traveling in the Mediterranean Sea*. Assume that country *A* is a non-EU country and has a list of foreign fighters from *A* and some of their biometrics data such as face images. Assume that *B* and *C* are EU countries and they have a shared list of foreign fighters from the EU. Assume also that the ship is from *B* and that according to the EU laws, the ship is authorized to make available the passenger list to analysts of *B* and *C* working on the identification of foreign fighters so they can do the matching and identify potentially suspicious passengers. However, the ship cannot provide the entire list of passengers to *A* as *A* is a non-EU country; the ship however can give *A* the list of passengers who are citizens of *A*.

Requirement 1: *Data fusion may be constrained based on the content of the data; that is, whether a dataset (which portion of a dataset) may be provided for a fusion task to a given subject depends on the content of the dataset (or portion of it).*

Requirement 2: *Data fusion may be executed only by subjects with specific roles and attributes.*

However, suppose that *A* wants to reduce the privacy "invasion" of its own citizens. Therefore, *A* requires receiving only the data of passengers that are citizens of *A* that have high probability of matching with the list of foreign fighters that *A* has.

Requirement 3: *Data fusion may have to be privacy-preserving with respect to its inputs.*

Finally, notice that results from this analysis must be made available to properly authorized officers at the arrival port of the ship for further questioning of the identified passengers by these officers.

Requirement 4: *Results of data fusion may be made available to subjects based on the roles and attributes of these subjects.*

Scenario 2. *Rescue ship with illegal immigrants traveling in the Mediterranean Sea*. Assume that a rescue ship has acquired some identity information from the rescued immigrants and is available to share this information with analysts working on identification of terrorists and returning fighters. This information, of course, cannot be trusted. Assume now that social networks have made available data about users from certain geographical areas to all three countries involved in the surveillance operations (e.g., *A*, *B*, and *C*). However, each country has in addition its own specific sources which are very sensitive and may include infiltrated agents and specialized equipment (such as very high precision reconnaissance UAVs and micro-aerial devices). Therefore, whereas all countries can fuse the data from the rescue ship with data from the social networks, each country can fuse such data only with data from its own sensitive sources.

Requirement 5: *Data fusion may be constrained based on the source of the data.*

Scenario 3. *Medical supplies for rescue ships.* Assume that rescue ships need to be provided with medical supplies and have to identify immigrants that may need to be quarantined. Intelligence agencies from countries *A*, *B*, and *C*, because of their sources, may have information about the areas where the immigrants are from. However, they may not want to disclose detailed data as that may give indications about their sources. They may thus only provide aggregate data (possibly perturbed) in order to maintain data confidentiality.

Requirement 6: *Data provided as input to fusion steps may have to be aggregated and/or sanitized.*

III. Background

We now introduce some background notions relevant to the discussion in the next sections.

A. Fusion Processes

Fusions processes can include multiple steps, such as preliminary steps for cleaning the data and data join steps in which different datasets are merged. We thus represent a data fusion process as a tree[1] where:

- Each leaf node denotes a dataset to be fused.
- Each non-leaf node denotes a data fusion step.

Note that fusion steps can be of different natures, including human analysis and use of various tools, such as machine learning techniques. An example of such a tree for Scenario 2 is given in Figure 1.

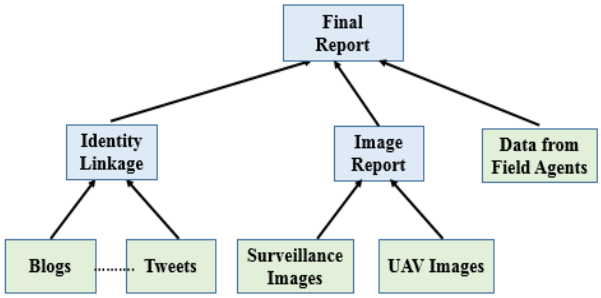

Figure 1. Fusion Tree for Scenario 2 (cfr. Section II)

B. Attribute-based Policy Models

Policy models typically include constructs for denoting the various entities involved in a given policy. For example, consider the case of an access control policy that specifies

[1] Actually a fusion process is more in general a graph as for example the result of a fusion step may be input to two different subsequent fusion steps. Here we assume that it is a tree in order to simplify the discussion.

which subject can access which data for which actions. Models for representing such a policy include constructs for denoting subjects and objects. Modern policy models are based on the notion of attributes, where an attribute represents a (security relevant) property of an entity of interest. Subjects and objects can thus be specified as Boolean combinations of predicates against such attributes [10].

Our policy model is based on such a notion. More specifically we assume that there are *roles* characterizing the subjects in the system, and *types* characterizing the protected resources in the system. Attributes for roles and types are defined in terms of metadata templates [8]. Such templates are pre-defined definitions of the attributes that are relevant for the policies. They define the domains for the attributes and how the attributes should be initialized and managed. The following definition introduces the notion of metadata template. It is a simplified version of the notion of metadata template from [8].

Definition 1. (**Metadata template**) Let OT be the set of object types and R be the set of roles existing in the system. A metadata template for an object type $ot_i \in OT$ or a role $r_j \in R$ is specified as follows.

MD-TEMPLATE *template-ID* **FOR** *target* {
    $attr_1$: dom_1;
    . . .
    $attr_n$: dom_n;
}

Where: *target* is either ot_i or r_j and represents the type (role) of the entity that is associated with the specified metadata template; $attr_i$, i =1, . . ., n, is the identification of the i-th attribute; and, dom_i, i = 1, . . . , n, is the domain for the i-th attribute. ◊

When a new object (subject) is introduced into a system, an instance of a metadata template is created for the object (subject), according to the metadata template specified for the type of the data (role of the subject). Then this metadata instance is associated with the object (subject) throughout its life-cycle in the system, that is, whenever some policy needs to be enforced, the metadata instances are retrieved in order to evaluate the conditions specified in the policy of interest. An important piece of information that can be included in the metadata for objects, which are datasets, is their provenance indicating among other information the source of the data [12].

IV. Policy Domains

As the scenarios in Section II illustrate, many different domain policies are required for a comprehensive control of security and privacy of data fusion processes. We classify these policies into three domains: access control policies, fusion policies, and derived data control policies. In addition, policies can have different applicability scopes*:* *local* to a fusion process, and *global*. An example of a policy with a global scope is a policy specifying that if one of the input datasets has top secret classification, the results of the data fusion is also top secret. Finally, policies can be *weak* or *strong* based on whether they allow exceptions or not [13].

In what follows we discuss in more details the different policy domains.

A. Access Control Policies

Access control policies have been widely investigated and many different models and mechanisms have been proposed. In our context we distinguish two types of access control policies: *basic policies*, and *separation-of-duty*/*binding-of-duty* (SoD/ BoD) policies. The former are the conventional policies in which the permission is based on the current access being requested. The latter constrain accesses based on past accesses executed by subjects. More specifically a SoD policy typically specifies constraints by which if a subject has executed a fusion step, the subject cannot execute another given fusion step. For example if a subject has generated an analysis report, the subject cannot validate the report; the validation must be done by another subject. By contrast, a BoD policy specifies that if a subject has executed a fusion step, the subject must execute some other given fusion steps. For example if a subject has validated a report, the subject must also validate the next versions of the report. In a way, BoD policies introduce obligations.

We now introduce an informal definition of the basic access control policy. We refer the reader to [14] for a formal model of SoD/BoD policies. In what follows we assume that each fusion process has a unique identifier taken from a set I of identifiers $\{i_1, i_2, \ldots, i_k\}$. In the notation adopted in the definition, the square brackets identify optional components of the policy, whereas the | symbol indicates alternatives.

Definition 2. (**Basic access control policy**). An access control policy P is an expression of the form:

P = < **weak** | **strong**, *Scp*, *S, D, A*>
Scp = **global** | **local** $\{i_1 [,i_2,\ldots, i_n]\}$
S = r [$\{pr\}$]
D = ot [$\{d\_pr\}$]
pr = $attr\ \phi\ v \mid pr \wedge pr \mid pr \vee pr \mid \neg pr$
d_pr = *pr* | *c_cnd*
A = **read** | **write** | **execute** ◊

The first component of a policy specification indicates whether the policy is strong or weak; the second component indicates whether the policy is global or local, and in the latter case, the fusion process(es) to which the policy applies. Finally, the third and fourth components are a subject specification and an object specification, expressed as a role specification optionally followed by a Boolean combination of predicates, and an object type specification followed by a Boolean combination of predicates, respectively. It is important to mention that, whereas for roles the predicates are against the role metadata (see Definition 1), for objects there are two types of predicates, namely: metadata predicates expressed against the object type metadata and content predicates. The latter are predicates against the content of the objects and are typically specified for objects that are datasets. The specific format of content predicates depends on the system managing the datasets. For example, if datasets are managed by a relational DBMS, the content predicates can be simply represented as SQL queries. In other cases, one may have to use tools to search contents of the datasets, and in these cases the condition is denoted by a function name.

Finally, we note that different types of actions can be specified in a policy; as which action can be executed on which object depends on the object type, a schema needs to be also provided with this information (see [13] for an example of such a schema).

Example 1. Consider Scenario 1 in Section II. Assume that the list of passengers is stored in a relational database as a table with name 'pass_list' and suppose that the identifier of the fusion process is *i*.

- The following policy specifies that analysts from an EU country can read the passenger list:
  < **strong**, **local**{*i*}, analyst {country ∈ {EU}}, pass_list, **read** >.
- The following policy specifies that analysts from country *A* can only read the list of passengers who are citizens of *A*.
  < **strong**, **local**{*i*}, analyst {country = A }, pass_list {Select * From pass_list Where citizenship = A}, **read** >.

Notice that among the protected objects we also include tools and systems that are actually used for executing fusion steps as use of these tools and systems may also be restricted. Moreover, as those tools and systems in most cases are given datasets as input, for a subject to run a tool (system) on one or more datasets, the subject must have both the authorization to execute the tool (e.g. **execute** in our informal grammar) and the authorization to read the input datasets.

B. Fusion Policies

Data fusion policies can be categorized into two types: (i) "data" separation policies, specifying essentially that two datasets cannot be fused together; and (ii) data pre-processing policies, specifying whether data must be sanitized and/or encrypted before being fused. In what follows we introduce an informal grammar for those policies. We assume that, in addition to the set *I* of identifiers of fusion processes, each fusion step, within a given fusion process, has an identifier from a set *IS* of identifiers $\{is_1, is_2, \ldots, is_k\}$.

Definition 3. (**Fusion policy**). A fusion policy *P* is an expression of the form:

P = < **weak** | **strong**, *Scp*, *DS* | *SN* | **mc** $\{is_i, D, D\}$ | $E(is_i)$ >
Scp = **global** | **local** $\{i_1 [i_2, \ldots, i_n]\}$
DS = $\{D, D\}$
SN = **anom** | **diffpriv** | **encr** (D)
D = ot [$\{d\_pr\}$]
d_pr = pr | c_cnd
pr = $attr\ \phi\ v$ | $pr \wedge pr$ | $pr \vee pr$ | $\neg pr$
E = **trustenv** | **enclave** ◊

The first two components in such a policy specification have the same meaning they have for the access control policies. The third component indicates the specific requirement for the policy. Such a requirement can be of four different forms: (i) data separation; (ii) data sanitization; (iii) privacy-preserving multiparty execution; and, (iv) execution environments. The first two forms of requirements apply to data; in particular a data separation constraint applies to two datasets (denoted possibly using conditions against their metadata) and specifies that these two datasets cannot be fused. The other data requirement specifies whether the dataset has to be sanitized (and whether by anonymization or by differential privacy techniques [15]) or encrypted. If a dataset has to be encrypted, the policy may also specify the type of encryption [16], the encryption algorithm, specific types of privacy-preserving protocols, and so forth. We omit these details from the grammar for simplicity. Finally the last two forms apply to the execution of the fusion step. They specify, respectively, that the fusion step must be executed according to some secure multiparty protocol (which depends on the type of function executed by the fusion step), and that the fusion step must be executed in a given execution environment (namely on a trusted platform or on a secure enclave). However, many other possible execution environments may be considered and added to list of environments that one can specify as part of the policies. A notable example is represented by protocols supporting the distributed privacy-preserving fusion of sensor data [17].

Example 2. Consider Scenario 1 in Section II. Assume that the list of passengers is stored in a relational database as a table with name 'pass_list'. In addition assume that the list of foreign fighters that *A* has is stored in a file with name 'fighter_list'. Finally suppose that the identifier of the fusion process step that performs the matching step for fusing passenger information for citizens of *A*, with A' information about foreign fighter is *is*.

- The following policy specifies that the fusion must be executed according to a multi-party protocol:
  < **strong**, **local**{*i*}, **mc** {*is*, pass_list {Select * From pass_list Where citizenship = A}, fighter_list} >.

C. Derived Data Control Policies

The derived data control policies can be categorized into two types: (i) policies specifying which subject can access the derived data – in this respect they are very much like the access control policies; and, (ii) policies that assign/modify the metadata attributes associated with the derived data. An example of the latter would be a policy specifying that if a dataset is obtained by merging two datasets, one of which is classified as "highly sensitive" (according to some sensitivity lattice), whereas the other is a publicly available dataset, the derived dataset has to be classified as "highly sensitive". However, there could be cases in which a "highly sensitive" derived dataset can be modified and downgraded. The derived data control policies allow one to support the various options. In addition, the derived data control policies may allow one to indicate the parties that maintain control on the derived datasets and thus administer the derived datasets. The parties controlling the derived datasets are also recorded as part of the metadata attributes associated with the derived datasets. We omit for brevity the information definition of these policies.

V. Policy Enforcement Techniques

Depending on the specific policy domain, different enforcement techniques need to be used. For access control policies, a reference enforcement architecture is the one proposed for XACML [18], which we report in Figure 2. Both the XACML language and the architecture provide features that can be used to implement fusion policies, including attribute conditions for both subjects and objects, as well as obligations that in our context can be used to implement

derived data control policies, for example to carry out actions to modify metadata associated with the derived data. In addition, attribute conditions may also include function calls by which one can wrap existing scripts and procedures. In addition, XACML is being integrated with relational DBMS so to support XACML policies for relational databases.

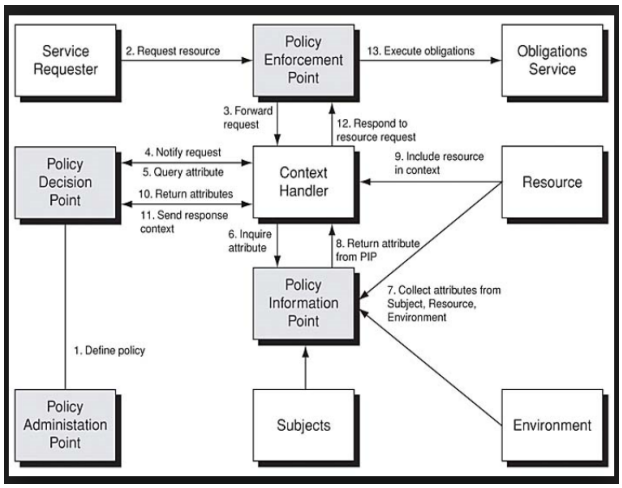

Figure 2. XACML Enforcement Flow Architecture [18]

The enforcement of fusion policies is more complex in that it may require the use of secure multi-party computation protocols, techniques for analytics on encrypted data, trusted environments, such as the Intel SGX platform, secure enclaves and classified environments. The main issue is to automatically determine the technologies to use based on the policies and which specific fusion tasks must be executed in which platform/environment.

An important aspect when dealing with data fusion policies is to make sure that policies do not conflict and their enforcement is feasible. For example, executing a fusion step that requires data to be ordered may not be possible for encrypted data. Techniques for policy analysis have been widely investigated mainly for access control policies and network management policies [19]. However, techniques for enforcement feasibility need to be devised. It is important to notice that the analysis of the enforcement feasibility may depend on the specific system and context in which the data fusion processes take place. For example, whereas in a centralized corporate-like environment, a secure enclave may easily be available, in an edge computing environment such an enclave may not be available to the user. Also in such environments, the fusion process may have a distributed execution – for example one device with special software may perform fusion on two image datasets, whereas another one may run a classifier on the fused dataset. Therefore, deciding which device to use for which fusion step must take into consideration the security and context of the device if there are policies requiring secure execution for such a step.

VI. Generative Policies

One of the challenges in specifying data fusion and data access control policies in coalition environments is the pace of change and dynamicity of the data sets that arise in these environments. The access control and data fusion policies syntax specified in the previous section require the identification of the data sets and their different attributes under which different conditional predicates can be specified and evaluated. As data fusion policies arise, and new data sources are encountered, the policies need to be explored and redefined to deal with the nuances arising from the current context and the attributes of the different data sets that are being encountered.

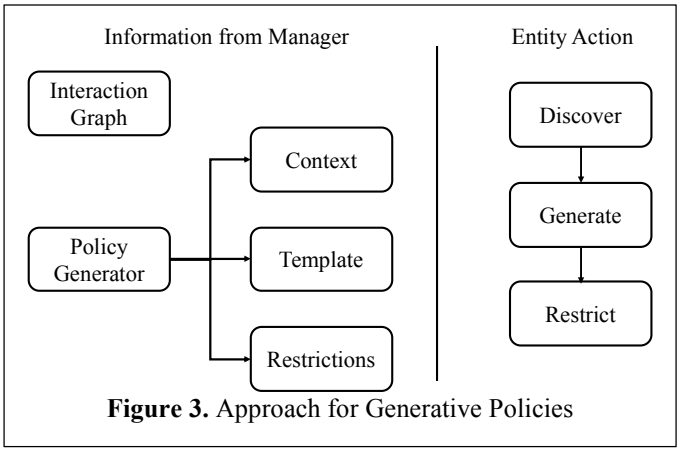

Figure 3. Approach for Generative Policies

An approach that holds promise for simplifying the policy specification process is the concept of generative policies [20], in which different entities (data sources and data fusion elements in our case) are given the freedom to generate their policies on their own, based on higher-level guidance given by a human administrator. The approach used is that of defining an interaction graph, which describes the different types of other entities that any entity may encounter in different contexts, along with a policy generator. The policy generator adds constraints on the type of policies that can be generated. Examining the definitions of the basic access control policy, data fusion policies and derived control policies in Section IV, the grammar provides additional constraints that augment the syntax defined in that Section.

The interaction graph defines for all entities (data sets or data fusion services) other types of entities they can expect in the system. The exact nature of entities depends on the type of data fusion operation being done. As an example, in the scenario of a refugee ship docking in different countries, the data set on the ship may need to interact with the social network sites available from different countries, and with the data fusion software available in different agencies (e.g., customs, police, navy and immigration) that is active in different countries. Different types of information would need to be given to each of these entities, depending on their role. The attributes on the nodes and links of the interaction graph identify which attributes are visible to each entity in the system (these define the attributes on the nodes) and which attributes are only visible to selected types of entities (these attributes appear on the links of the graph).

The policy generator adds constraints on the type of policies that are to be generated by each entity for exporting or ingesting data, depending on the context of the data fusion.

The generator can be viewed as having three main components: a definition of the context, a definition of the policy template or syntax, and any restrictions on the policy being generated. In many cases, these can be expressed succinctly as a context free grammar. In some cases the template may specify an entire set of policies, and the entity only needs to fill in the attributes of the different discovered entities in the system. In other cases, the restrictions may provide some limits on the different values fields can take. The entity can run an optimization algorithm or learning algorithm to come up with a set of viable policies, and then choose a subset that conforms to the restrictions that are provided. The data set on the ship may decide, when exporting data, that it is more efficient for it to only export the minimum information required by the agency of the country when it is interfacing with a fusion element in the immigration agency of the country. On the other hand, when interfacing with the law enforcement agency fusion element, it may decide to export all the information that is required because the low enforcement agency is running the operation in a secure enclave which is more trusted.

The types of information provided to the managed device and the actions of the device are shown in Figure 3. The left hand side of the diagram shows the information provided by the manager as described above, and the right hand side shows the action of the device/entity. The device/entity discovers other devices/entities in the system. The states of the different discovered devices define the context of the system. The device then tries to generate a set of policies suitable for the context. It can use the templates from the manager, use a learning algorithm, or some type of optimization to determine its policies. The resulting policies need to be restricted to a subset that can conform to the restrictions provided by the manager.

One of the ways in which an entity may determine its policies is by using learning from prior examples. As a simple illustration of this approach, let us consider the refugee ship when it docks in the first EU country. The exact set of access control, data fusion and distributed control policies may need to be defined manually. Now, let us assume the ship docks in another EU country. If the interaction graph identifies both these EU country agencies in the same role, the ship can use the same policies (with appropriate changes in attributes) for the other country. As more examples are collected, a learning process can both learn new processes, as well as refine the interaction graph (e.g., prune off roles that are never exercised, or modify attributes depending on the corrections made by the human administrator on the ship).

VII. Concluding Remarks

In this paper, we have identified initial requirements and policy models, and policy domains for data fusion processes. The focus of the policies is to ensure security and privacy.

As future work, we will investigate how to apply the generative policy model to derived data. The derived data control policies would seem to benefit most from automatic generation. When a fusion process creates new data elements, sensitive information can be revealed, both about the data that has been fused and the fusion function. The attributes of the derived data would depend upon the attributes of the data sources that were used in the derivation, and the attributes of the fusion process that created the derived data. The number of alternatives could thus be very large, and specifying policies for every conceivable situation would be onerous. The access control policies and the fusion policies for the derived data would ideally be created automatically.

Additionally, we plan to investigate other categories of policies that may be relevant for data fusion. An important direction for our future work is to extend the proposed models with ontological representations so to be able to organize relevant concepts, such as user roles and data types into ontologies so to be able to support inferences. We also plan to extend our policy models with the notion of contexts that would also be characterized by attributes and possibly organized according to an ontology. Last but not least we plan a prototype implementation of our policy system using a workflow management system to represent fusion processes and the XACML access control system, extended with ontological reasoning (along the lines of [21]).

References

[1] M. Pasca, "Weakly-supervised Discovery of Named Entities Using Web Search Queries", Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007.

[2] X. Yu, Y. Sun, E. Bertino, X.Li, "Modeling User Intrinsic Characteristic on Social Media for Identity Linkage", Proceedings of the 2018 ACM Conference on Supporting Groupwork, GROUP 2018, Sanibel Island, FL, USA, January 07 - 10, 2018.

[3] M. Rezvani, A.Ignjatovic, E. Bertino, S. K. Jha, "Secure Data Aggregation Technique for Wireless Sensor Networks in the Presence of Collusion Attacks", IEEE Trans. Dependable Sec. Comput. 12(1): 98-110 (2015).

[4] J. Bleiholder, F. Nauman, "Data Fusion", ACM Comput. Surv. 41(1):1-1:41 (2009).

[5] M.Haghighat, M.Abdel-Mottaleb, W.Alhalabi, "Discriminant Correlation Analysis: Real-Time Feature Level Fusion for Multimodal Biometric Recognition", IEEE Trans. Information Forensics and Security 11(9): 1984-1996 (2016).

[6] L. Sweeney, "K-Anonymity: a Model for Protecting Privacy", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5): 557-570 (2002).

[7] F.Y. Rao, J.Cao, E. Bertino, M. Kantarcioglu, "A Hybrid Private Record Linkage Scheme: Separating Differentially Private Synopses from Matching Records", Proceedings of the 31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015.

[8] J.-W. Byun, Y. Sohn, E. Bertino, "Systematic Control and Management of Data Integrity", Proceedings of the 11th ACM Symposium on Access Control Models and Technologies, SACMAT 2006, Lake Tahoe, California, USA, June 7-9, 2006.

[9] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, "Edge Computing: Vision and Challenges", IEEE Internet of Things Journal 3(5):637-646 (2016).

[10] E. Bertino, G. Ghinita, A. Kamra, "Access Control for Databases: Concepts and Systems", Foundations and Trends in Database 3(1-2):1-148 (2011).

[11] IETF, "Policy Core Information Model (PCIM) Extensions", January2003, available at https://tools.ietf.org/html/rfc3460.

[12] A.A. Jabal. E. Bertino, "SimP: Secure Interoperable Multi-granular Provenance Framework", Proceedings of the 12th IEEE International Conference on e-Science, e-Science 2016, Baltimore, MD, USA, October 23-27, 2016.

[13] F.Rabitti, E. Bertino, W. Kim, D. Woelk, "A Model of Authorization for Next-Generation Database Systems", ACM Trans. Database Syst. 16(1): 88-131 (1991).

[14] E. Bertino, E. Ferrari, V. Atluri, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. ACM Trans. Inf. Syst. Secur. 2(1): 65-104 (1999).

[15] C. Dwork, "Differential Privacy", Proceedings of the 33rd international conference on Automata, Languages and Programming, ICALP 2016, Venice, Italy, July 10-14, 2006.

[16] R.L. Lagendijk, Z. Erkin, M. Barni, "Encrypted Signal Processing for Privacy Protection: Conveying the Utility of Homomorphic Encryption and Multiparty Computation", IEEE Signal Process. Mag. 30(1): 82-105 (2013).

[17] M. Ambrosin, P.Braca, M. Conti, R.Lazzeretti, "ODIN: Obfuscation-Based Privacy-Preserving Consensus Algorithm for Decentralized Information Fusion in Smart Device Networks", ACM Trans. Internet Techn. 18(1): 6:1-6:22 (2017).

[18] XACML. Extensible Access Control Markup Language (XACML) Version 3.0. http://docs.oasis-open.org/xacml/3.0/xacml-3. 0-core-spec-os-en.html [Last Accessed: Jan. 27, 2018].

[19] A.A. Jabal et al., "Methods and Tools for Policy Analysis", submitted for publication, February 2018.

[20] E. Bertino, S. Calo, M. Touma, D. Verma, C. Williams & B. Rivera, B. "A Cognitive Policy Framework for Next-Generation Distributed Federated Systems: Concepts and Research Directions," Proceedings of IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, Atlanta, GA, June 2017.

[21] R. Ferrini, E. Bertino, "Supporting RBAC with XACML+OWL", Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT 2009, Stresa, Italy, June 3-5, 2009.