Modelling of Integrated Trust, Governance and Access

safi.re: Information Sharing Architecture

William J. Buchanan[1(✉)], Omair Uthmani[1], Lu Fan[1], Niall Burns[1], Owen Lo[1], Alistair Lawson[1], James Varga[2], and Cassie Anderson[2]

[1]School of Computing, Edinburgh Napier University, 10 Colinton Road, Edinburgh EH10 5DT, UK
w.buchanan@napier.ac.uk
[2]miiCard, Elliot House, 8 Hillside Crescent, Edinburgh EH7 5EA, UK

Abstract. We live in a world where trust relationships are becoming ever more important. The paper defines a novel modelling system of trust relationships using Binary Decision Diagrams (BDDs), and outlines how this integrates with an information sharing architecture known as safi.re (Structured Analysis and Filtering Engine). This architecture has been used on a number of information sharing projects, including within health and social care integration, and in sharing between the police and their community partners. The research aims to abstract the relationships between domains, organisations and units, into a formal definition, and then implement these as governance rules, and using the trust relationship definition, and the rules.

Keywords: Information sharing · Trust · Governance · Binary Decision Diagrams

1 Introduction

In an increasingly connected world, data is becoming a key asset, especially within a Big Data context, where data from different domains can be brought together to provide new in-sights. Most of the systems we have in-place, though, have been built to securely keep data behind highly secure environments, and then have difficulty in integrating with other disparate systems. This is now a major barrier to using data in a wide range of applications. Along with this, information sharing has many regulatory constraints, which often disable information sharing across domains, but, with carefully managed information architectures, it is possible to overcome many of these problems. An important challenge is thus to support information sharing across different domains and groups, across multiple information systems. In the context of this paper, a domain is defined as the governance (and possible ownership) of a set of data, which is exposed to others through well-managed services.

The problem of providing governance around trusted infrastructures is highlighted by Boris Evelson who outlines that [6]:

M. Felici (Ed.): CSP EU FORUM 2013, CCIS 182, pp. 91–101, 2013.
DOI: 10.1007/978-3-642-41205-9_8,
© Springer-Verlag Berlin Heidelberg 2013

> Big data is such a new area that nobody has developed governance procedures and policies, there are more questions than answers.

A feature of any trusted infrastructure is that the owner of the data is clearly defined, and this entity can differ from the actual governance of it. For example, in a health care system, the owner of the data can be the citizen, and the governance of the data is defined by the health care provider (such as the National Health Service (NHS) in the UK). In a full trust infrastructure, the citizen could have full rights to define who had access to their data.

The safi.re architecture has been used in a number of projects including within health and social care, including with the TSB-funded project with Chelsea and Westminster Hospital in London which focused on creating an e-Health Cloud within a hospital environment [1]. This used a novel method of defining the ownership of the data, and providing a rights infrastructure for the citizen (or patient) to define the rights of access to their data. This work has since been extended within a number of projects including the TSB Trusted Service project, which has focused on integrating both digital and human trust, to provide a fully integrated and holistic care infrastructure, and which integrates primary and secondary health care with assisted living [2–4].

Another important area for information sharing is within the holistic care, where information from different public sector agencies can be used to improve the care of citizens. This might relate to sharing information on a child for concerns posted within health, social care, education and policing, where concerns within just one of these domains would not be seen as a major concern, but when aggregated across several of these, it might result in the concerns being escalated to the point where an action plan is initiated [5]. The work has thus into projects which involve information sharing for Child Protection, and which integrate with a multi-agency approach. As there is information held within each of the public sector agencies, it is important that accesses are well managed and controlled for the rights for the access to data.

The next section outlines the safi.re information sharing and how information rules are defined. After this the paper outlines a novel method for the modelling of information sharing using Binary Decision Diagrams (BDDs), and show how this integrates with the architecture. The final sections outline the results of simulations and in the main conclusions of the work.

2 Architecture Outline

A major problem within many information infrastructures is the control of information between organisational boundaries. Normally this is defined with a security policy, but the scope of this is often just defined within an organisation. As more information crosses organisations and domain boundaries, it is becoming difficult to manage the number of possible ways that information can be shared and aggregated. A key element of this is the increasing requirement for trust between organisations and units, especially with the move towards cloud-based services. The safi.re (Structured Analysis, Filtering and Integrated Rules Engine) architecture overcomes these

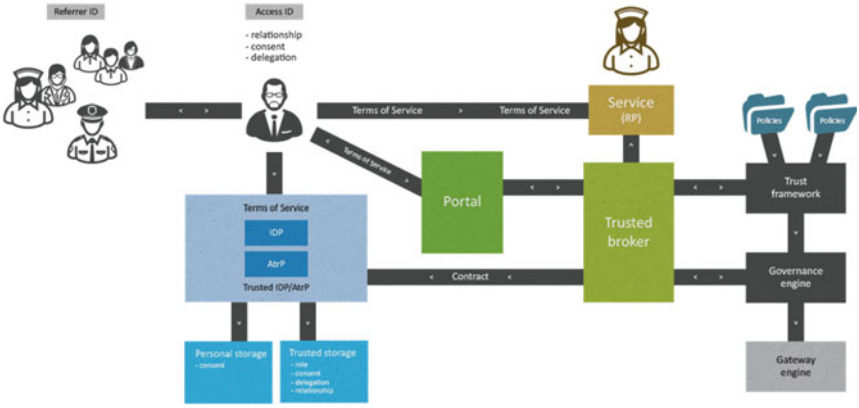

Fig. 1. Trust, governance and access framework

problems by creating a formal structure for the abstraction, governance and implementation of trust relationships and security policies. It can be used as a full end-to-end solution for policy abstraction, implementation and controlled access to services, or can integrate each of the elements as a Service to existing applications.

safi.re supports three basic components, each of which can operate as a stand-alone product or can integrate with existing systems. These are:

- **Safi.re TRUST**. This is a trust framework which abstracts the roles and services and defines their trust relationship. The export from this component is the requirements for the information sharing/service aggregation policy.
- **Safi.re GOVERNANCE**. This takes, as an input, the abstraction of the trust framework, and runs the rules required to define if an entity has the rights to access a given service.
- **Safi.re GATEWAY**. This takes the rules from the governance engine, and implements them within a real-time filtering system, which controls all the accesses to services between the domains.

In modern service-oriented infrastructures a user must gather claims to consume a service. Too often the service is bound to a specific authentication infrastructure which limits the scalability of the provision of the service. For more dynamic infrastructures there is no direct communication between the service and the gathering of the claims around identity and the attributes required to consume a service. Figure 1 thus outlines this process, where there are Terms of Service (ToS) between a user and their identity and attribute provider, another ToS between them and the service, and so on. It is the focus of the Trust and Governance infrastructure to define a contract which binds these terms of service together. This contract pre-defines the requirements for the claims to the service, and then is trusted to actually issue the contract for the user to consume the service.

3 Trust, Governance and Access

This paper is based on the integration of a formal trust framework and implemented rules, and then modelling of complex trust relationship between domains using a patent pending method of Binary Decision Diagrams (BDDs) [14]. BDDs are rooted, directed, acyclic graphs originally proposed by Lee [7] in 1959 and Akers [8] in 1978 to graphically represent Boolean functions. BDDs originate from binary decision trees which are rooted, directed trees that can be used to represent Boolean functions. For example, the decision tree illustrated in Fig. 2 represents the Boolean function f(x; y) = (x ∨ y).

3.1 Reduced Ordered Binary Decision Diagrams (ROBDDs)

In 1986, Randal Bryant proposed a solution to this problem in [9] by introducing algorithms for reducing binary trees and ordering the variables in a function. The process of reduction consists of merging any isomorphic sub-graphs for the decision tree. Any parent node which has child-nodes that are isomorphic is considered redundant and is removed. Applying this process to the decision tree for the Boolean

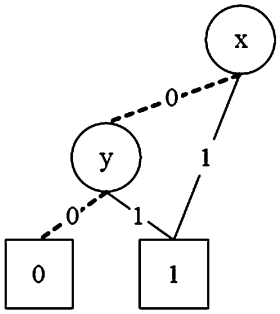

Fig. 2. Reduced Binary Decision Diagram for the function f(x; y) = (x ∨ y)

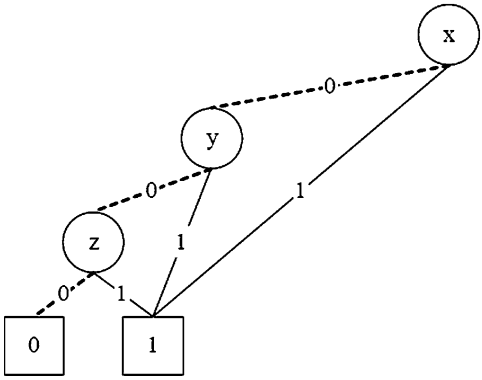

Fig. 3. Reduced Binary Decision Diagram for the function f(x; y; z) = (x ∨ y ∨ z)

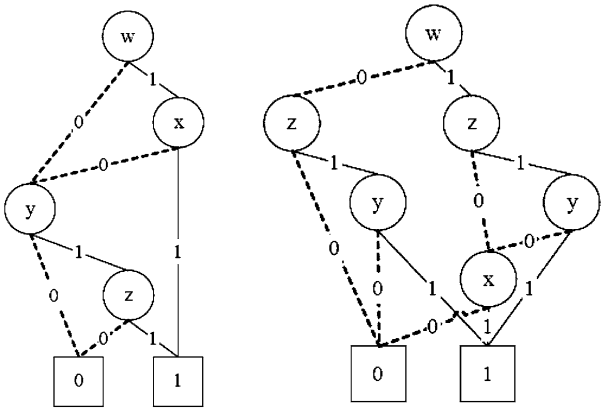

Fig. 4. Reduced Binary Decision Diagram for the function f(w; x; y; z) = (w ∧ x) ∨ (y ∧ z) with variable ordering of w; z; y; x

function f(x; y) = (x ∨ y), as illustrated in Fig. 2, it is evident that if the first node, x, is 1, then the value of the second node, y, has no effect on the terminal node value of the Boolean function: whether y is 0 or 1, the value of the terminal nodes is 1. This means that the where node x is 1, child-nodes of y are isomorphic. Node y can then be considered redundant here and removed. The result is the reduced decision tree illustrated in Fig. 3. Similarly, applying the reduction process to the decision tree for the Boolean function f(x; y; z) = (x ∨ y ∨ z), illustrated in Fig. 3, yields the reduced decision tree shown in Fig. 4. Reduced decision trees allow a much more compact representation of Boolean expressions than non-reduced decision trees.

Bryant also highlighted in [9] that the size of a decision tree for a given function is dependent on the ordering of the variables in that decision tree. For example, the decision tree for the Boolean function f(w; x; y; z) = (w ∧ x) ∨ (y ∧ z), given a variable ordering of w; x; y; z, is illustrated on the left-hand diagram in Fig. 4.

If the variable ordering for the same function was now changed to w; z; y; x, the resultant decision tree will be more complicated, as illustrated in the right hand side of Fig. 4. Hence, an optimal variable ordering will produce the simplest, and therefore smallest, decision tree for a given function, while sub-optimal orderings will produce larger and more complex decision trees for the same function. However, as shown by Bollig and Wegener in [10], determining the optimal variable ordering for a Boolean function is an NP-complete problem that often requires trial and error or expert knowledge of domain-specific ordering strategies.

Decision trees which have been reduced and ordered are referred to as Reduced Ordered Binary Decision Diagrams (ROBDDs), or commonly shortened to just Binary Decision Diagrams (BDDs). A key property of the reduction and ordering restrictions introduced by Bryant is that the resulting BDDs are canonical [13]. This means that the BDD for any Boolean function, for a defined variable ordering, will always be isomorphic. This property has made BDDs ideal for use in formal equivalence checking. In the electronic design automation process, for example, BDDs are frequently used to formally prove that two circuit design representations exhibit the same behaviour.

3.2 BDDs in Policy Modelling

A novelty of this paper is to exploit the unique properties of Binary Decision Diagrams (BDD) to model complex sets of policies, in a form that is readily machine-executable, and to extend these to the information-sharing domain. The work of Hazelhurst et al. [11] with firewalls identified key constituent fields in access-list rules and translated these into bit vectors representing BDD variables. This research applies a similar methodology to information-sharing where a set of information-sharing policies can be modelled as a decision diagram, once a specific variable ordering scheme has been selected. The modelling of a set of policies as a BDD provides a number of significant advantages, including providing an efficient lookup mechanism for an information-sharing request as well as providing a graphical representation of the overall policy set. As rule sets become larger and more complex, they become difficult to interpret and maintain [12]. Modification of the rule set, by either adding new rules or removing existing ones, or even changing the order of rules has a significant impact on the behaviour of the policy-based system. Hence, analysis and validation of large, complex rule sets is essential in ensuring that high-level directives are enforced. Further, exploiting the formal equivalence checking ability of BDDs, and the fact that they can canonically represent Boolean functions, multiple sets of policies can be compared to ensure that they have the same behaviour or identify areas where they behave differently. Large and complex rule sets, represented as BDDs, can, therefore, be efficiently modelled, analysed and validated.

3.3 Domain Modelling Using BDDs

The core of the patent is the linkage with the trust framework and the governance rules. In order to simply the access to data from domains, the method exposes only well-managed services to define the trust relationship. Within this the model defines a number of modelling elements, including:

- **Permission**. This is a simple permit or deny for access to a service.
- **Domain**. This relates to the domain that an accessor is contained within, and is used to create the holder to the domain ontology.
- **Organisation**. This relates to an organisation with a given domain.
- **Unit**. This relates to a unit with an organisation.
- **Role**. This defines the role that an accessor has in access a service within another domain.
- **Relationship**. This defines the relationship that the accessor has to the data being accessed.
- **Action**. This defines a CRUD (Create, Read, Update or Delete) access to a service and its associated data.
- **Attribute**. This defines an attribute of the object to be access, such as for a health record.
- **Object**. This defines the actual access target, such as for a specific person.
- **Context**. This defines the content of the investigation (which can be used to define certain risk levels for access privilege escalation.

- **Compliance**. This defines the audit/compliance reasons for the access.

The trust framework then defines the usage of each of these fields, and rules are written which implements them. A sample rule is thus:

> [Permit] [Police.Police_Force_A.*.Sergeant] with [*] relationship [R] [Unique_Identifier] of [Child] with [Abuse_Investigation] context from [Social_Care.Child_Protection_Agency_B.Records_Unit.Records_Admin] with Compliance [Human_Rights_Act_1998]

Overall the BDD model uses a binary representation for each of these fields, and which builds-up a rule definition with the binary representation of each of the possibilities for the fields. For example if there are four roles, we can represent them with:

- 00 – Constable
- 01 – Sargent
- 10 –Superintendent
- 11 – Chief Superintendent

These rules then use the BDD to determine if there are issues within the governance rules related to:

- **Redundancy**. This is where one set of rules is already included within the trust rules already defined.
- **Shadowing**. This is where a rule is higher up in the set of rules, and matches all the conditions that match in the current rule, such that the shadowed rule will never be activated.
- **Generalisation**. With this a rule is generalisation of another preceding rule if it matches all the packets of the preceding rule.
- **Correlation**. Two rules are correlated if the first rule in order matches some of the fields of the condition of the second rule and the second rule matches some of the fields of the condition of the first rule.

3.3.1 Simple Example

This example describes in detail the steps needed to translate a set of information-sharing policies to a BDD. List 1 shows a sample list of policies. A '*' or 'Any' is used to denote redundant fields, or redundant portions of fields. Redundant fields are not translated into binary as they represent variables that are not evaluated by the BDD and, hence, do not form part of the Boolean function. Where an entire field is redundant, it is entirely excluded from the binary representation and where only a portion of a field is redundant, only the relevant portion is translated while the redundant portions are shown using 'Xs'.

Listing 1:

Policy 1: This policy <permits> <ANY> requester, with <ANY> relation in <ANY> context, to request to <read> a <child's> <Health History Record> from the <Records Admin> of the <Records Unit> of <Child Protection Agency 'B'> under compliance of the <Data Protection Act>

Policy 1:

Compliance (DPA)	: 1
Requester (Any)	: not checked by BDD
Relation (Any)	: not checked by BDD
Context (Any)	: not checked by BDD
Object (Child)	: 1
Attribute (Health History Record)	: 01
Owner (SocCare.CPA-B.RecUnit.RecAdmin)	: SocCare : 10
	: CPA-B: 10
	: RecUnit : 10
	: RecAdmin : 10
Action (Permit)	: 1

The Boolean function corresponding to Policy1, ignoring redundant fields, is a logical conjunction of all of the above fields in the format shown in Listing 2. Listing 2 represents Policy1 expressed logically as an 'if–then' conditional statement and Fig. 1 illustrates Policy1 as a BDD.

Listing 2:

Permit: Compliance ^ Owner ^ Object ^ Attribute

Listing 3: Rule1 expressed as an if–then conditional statement.

```
if      (Compliance = 1) ^
        (Owner = 10101010) ^
        (Object = 1) ^
        (Attribute = 01),
then (Action = Permit)
```

4 Results

This section offers an overview of the total processing times for a 2.6 GHz processor with 2 GB of memory, using a range of policy-set sizes. Two initial tests comprising of 1,000 policies and 10,000 are run in order to ensure that the test platform is stable. Following these, tests are run starting with a set of 100,000 policies and repeated at increments of 50,000 policies, until a maximum set of 1,000,000 policies is tested. During each test, measurements of percentage CPU utilisation, RAM usage and

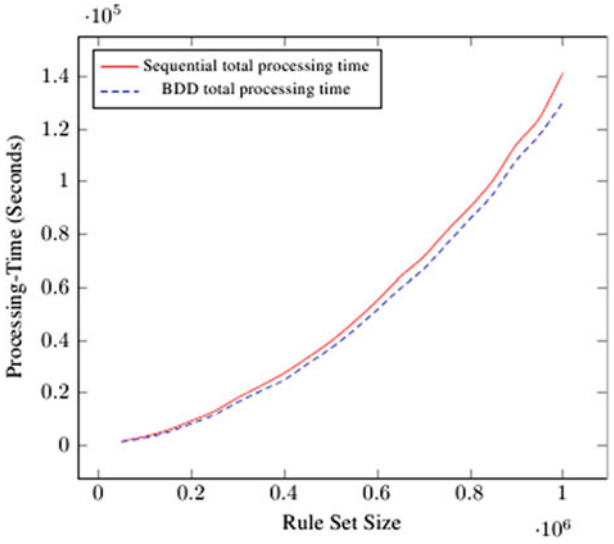

Fig. 5. Comparison of total processing times using sequential and BDD methods against increasing rule set size

latency are gathered for each stage of the policy verification process. Further, measurements for the anomaly detection stage are gathered for both sequential and BDD modes of operation.

As illustrated in Fig. 5, the total processing times increase with respect to increasing policy set size. Hence, with the same available resource configuration, it is expected that the total processing time will increase proportionally to the size of the policy set. In fact, as illustrated by the graphs in the figures mentioned, the rate of change of the total processing time increases with increasing policy set size, indicating a polynomial relationship.

A related observation from these is that the total processing times for the sequential method are higher than the total processing times for the method using Binary Decision Diagrams (BDDs). This result is expected, as the process using BDDs, due to their tree structure, involves fewer computations than sequential comparisons. Further, it should be noted that the total processing time for the sequential process increases at a greater rate than the total processing time for the process using BDDs.

5 Conclusions

This paper has outlined the safi.re architecture which integrates a trust framework and governance rules. As information opportunities increase with a sharing across domains, the modelling of polices is becoming important especially in identifying redundancy, shadowing, generalisation and correlation. The method defined in this

paper uses BDDs, which support a structured approach to the modelling, and which identifies problems in the governance rules.

As trust relationships are becoming a key focus within defining the security infrastructure between organisations, the complexity of these relationships is becoming a key factor. If we simplify these too much, it reduces the problem to simplistic rules which often do not reflect the actual inter-relationship between organisations. The long term goal must thus be to implement a trust infrastructure which can properly define the interaction between organisations, and this will require large-scale modelling of these. A key element of this will be the modelling of these governance rules between organisations as these will identify problems in the organisation of the governance rules. The BDD method outlined in this paper thus supports the next generation of trust relationships, and their related governance rules, and provides a method to reduce complexity of these.

While the BDD method has been applied to static modelling of policies such as in modelling network firewall rules, there is a need to model temporal rules, thus the modelling requires to be undertaken at frequent intervals in order to catch new rules which may conflict with existing ones.

References

1. Fan, L., Buchanan, W., Thuemmler, C., Lo, O., Khedim, A., Uthmani, O., Lawson, A., Bell, D.: DACAR platform for eHealth services cloud. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 219–226 (2011)
2. Fan, L., et al.: SPoC: protecting patient privacy for e-Health services in the cloud. In: The Fourth International Conference on eHealth, Telemedicine, and Social Medicine, eTELEMED 2012 (2012)
3. Ekonomou, E., Fan, L., Buchanan, W., Thuemmler, C.: An integrated cloud-based healthcare infrastructure. In: 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 532–536. IEEE, November 2011
4. Lo, O., Fan, L., Buchanan, W.J., Thuemmler, C.: Technical evaluation of an e-health platform. IADIS E-Health (2012)
5. US Patent Application No 13/739074, The Court of Edinburgh Napier University, Short Title: Binary Decision Diagrams, IP Title: Improved Information Sharing, Submitted: 11 Jan 2013
6. Roger du Mars, Mission Impossible? Data Governance takes on Big Data, Boris Evelson, BI Trends and Strategies (2012). http://cdn.ttgtmedia.com/searchBusinessAnalytics/downloads/BI_Trends_+_Strategies_July_2012.pdf. Accessed July 2012
7. Lee, C.: Representation of switching circuits by binary decision programs. Bell Syst. Tech. J. **38**, 985–999 (1959)
8. Akers, S.B.: Binary decision diagrams. IEEE Trans. Comput. **C-27**(6), 509 (1978)
9. Bryant, R.: Graph-based algorithms for boolean function manipulation. IEEE Trans. Comput. **C-35**(8), 677–691 (1986)
10. Bollig, B., Wegener, I.: Improving the variable ordering of OBDDs is NP-complete. IEEE Trans. Comput. **45**, 993–1002 (1996)

11. Hazelhurst, S., Fatti, A., Henwood, A.: Binary decision diagram representations of firewall and router access lists. University of the Witwatersrand, Johannesburg, South Africa, Tech. Rep. TR-Wits-CS-1998-3 (1998)
12. Hazelhurst, S.: Algorithms for analysing firewall and router access lists. University of the Witwatersrand, Johannesburg, South Africa, Tech. Rep. TR-WitsCS -1999-5 (2000)
13. Bryant, R.E.: Symbolic boolean manipulation with ordered binary-decision diagrams. ACM Comput. Surv. **24**, 293–318 (1992)
14. US Patent Application: 13/739074
15. Hardt, D. (ed.): The OAuth 2.0 Authorization Framework, IEFT RFC 6749. http://tools.ietf.org/html/rfc6749.html (October 2012)