Dynamic Labelling to Enforce Conformance of Cross Domain Security/Privacy Policies

N.V. Narendra Kumar(✉) and R.K. Shyamasundar

Department of Computer Science and Engineering,
Indian Institute of Technology Bombay, Mumbai, India
naren.nelabhotla@gmail.com, shyamasundar@gmail.com

Abstract. Conformance of declared security policies while traversing different sites has been a challenge for realizing work-flows on clouds that need to move from one cloud domain to another domain from the perspective of optimization of utilization. Such a possibility will enable optimization of communication and thereby realize the tenet of Utility Computing or Cloud computing. In this paper, we show how dynamic relabelling realized through the Readers-Writers Flow Model (RWFM) enables us to realize such an important property. We shall illustrate the modelling through an example wherein the privacy policies of two domains that permit each other have different security policies and show how, it is possible to realize a joint policy that is in conformance with both the domains. This enables us to inform the user when the privacy policy for which he has signed differs from the cross-site traversal and thereby assure him that his main privacy policy is preserved. If not, he can provide an explicit endorsement as long as that will not compromise the security policy of the main domain for which he has signed.

Keywords: Privacy · Information flow · Conformance

1 Introduction

Current web applications often present a single visualization of resources accessed from multiple domains. This is enabled by mashups using the technology of cross-origin resource sharing (CORS). While this presents a large potential for business development, it also brings with it serious security challenges. In this context, it becomes imperative to have strong access controls to prevent devastating attacks including cross-site request forgery (CSRF) and cross-site scripting (XSS) which have been consistently featuring in the OWASP top 10 list of web vulnerabilities. In particular, even accidental disclosure of private data (data consisting of personal identifiable information) could lead to a crippling/life-altering effect on the user whose privacy has been compromised.

N.V. Narendra Kumar—The work was carried out with support from ISRDC (Information Security Research and Development Center), a project sponsored by MeitY, GoI.

P. Krishnan et al. (Eds.): ICDCIT 2017, LNCS 10109, pp. 183–195, 2017.
DOI: 10.1007/978-3-319-50472-8_15

Privacy is quite a complicated notion in this digital world. In general terms, privacy has been defined to be the ability of an individual or group to seclude themselves, or information about themselves, and thereby express themselves selectively. Formalization of privacy has gained significant importance in the context of querying big data, health information systems, electronic patient records, etc. [2]. Consider the classic example of interactions between a patient and a physician discussed in [2]:

- A patient goes to a hospital and provides medical information to a physician.
- The physician needs help and passes some of this information to a nurse.
- The nurse then turns around and sells the information to a drug company that uses it for marketing.

Since some patients object to such uses of their health information and the society, wants to encourage open communication between patients and their physicians, countries like USA have adopted privacy policies, such as HIPAA, that prohibit such uses of patient information without their consent. To ensure that employees comply with these policies, hospitals employ auditors who examine accesses to and transmissions of protected information looking for actions that violate the privacy policies in place.

In view of this, a new research area of formalizing privacy policies has emerged [2] with the following two broad objectives:

1. Formalize Privacy Policies: Arrive at precise semantics of privacy concepts,
2. Enforce Privacy Policies for purposes like:
   - Audit: Detect violations of policy, and
   - Accountability: Identify agents to blame for policy violations and punish to deter policy violations.

One of the widely used strategies [2] is to formalize the notion of *purpose* in privacy policies. For instance,

- Yahoo!s privacy policy governing its email service says that it "will not use the contents of emails for marketing purposes".
- The social security administrations privacy policy says it "will use information collected only for the purposes for which it was collected".

Possible interpretation of the above examples are as envisaged in [2]:

- Yahoo!'s practice is not to use the content of messages [...] for marketing purposes. That is: Yahoo!s policy is an example of an *"not-for" restriction*,
- By providing your personal information, you give [Social Security Administration] consent to use the information only for the purpose for which it was collected. This will be an example of *"only-for" restriction*

Purpose restrictions are widely used in specifying privacy policies. Some of the prominent ones are mentioned below:

– OECDs Privacy Guidelines,
– US Privacy Laws such as HIPAA, GLBA, FERPA, COPPA ...
– EU Privacy Directive
– Enterprise Privacy Policies Google, Facebook, Yahoo ...

In this paper we demonstrate that "not-for" and "only-for" purpose restrictions can be succinctly captured through the dynamic labelling of `RWFM` discussed in Sect. 2 and further, the `RWFM` model permits a wide range of specifications.

The rest of the paper is organized as follows: a brief introduction to our Readers-Writers Flow Model is presented in Sect. 2, and its application to verifying and enforcing compliance of privacy policies across multiple interacting domains is described and illustrated with an example in Sect. 3. Section 4 discusses the application of our technique for protecting hybrid cloud environments, and Sect. 5 provides comparison with related work and concluding remarks.

2 Readers-Writers Flow Model (RWFM)

The Readers-Writers Flow Model (RWFM) proposed in [4,5] is a novel model for information flow control. RWFM is obtained by recasting the Denning's label model [3], and has a label structure that: (i) explicitly captures the readers and writers of information, (ii) makes the semantics of labels explicit, and (iii) immediately provides an intuition for its position in the lattice flow policy.

Recasting Procedure. Given a Denning's lattice model $DFM = (S, O, SC, \oplus, \leqslant)$ with flow policy $\lambda : S \cup O \rightarrow SC$, we recast the labels in terms of the readers and writers to obtain an equivalent flow policy defined by $DFM_1 = (S, O, SC_1, \oplus_1, \leqslant_1)$ and $\lambda_1 : S \cup O \rightarrow SC_1$, where:

(i) $SC_1 = 2^S \times 2^S$,
(ii) $\oplus_1 = (\cap, \cup)$,
(iii) $\leqslant_1 = (\supseteq, \subseteq)$, and
(iv) $\lambda_1(e) = (\{s \in S \mid \lambda(e) \leqslant \lambda(s)\}, \{s \in S \mid \lambda(s) \leqslant \lambda(e)\})$, where e is a subject or object.

RWFM is obtained by generalizing the above recasting procedure, and is defined as follows:

Definition 1 (Readers-Writers Flow Model (RWFM)). *Readers-writers flow model RWFM is defined as the 8-tuple* $(S, O, SC, \leqslant, \oplus, \otimes, \top, \bot)$*, where*

S *and* O *are the set of subjects and objects in the information system,*
$SC = S \times 2^S \times 2^S$ *is the set of labels,*
$\leqslant = (-, \supseteq, \subseteq)$ *is the permissible flows ordering,*
$\oplus = (-, \cap, \cup)$ *and* $\otimes = (-, \cup, \cap)$ *are the join and meet operators respectively, and*
$\top = (-, \emptyset, S)$ *and* $\bot = (-, S, \emptyset)$ *are respectively the maximum and minimum elements in the lattice.*

The first component of a security label in RWFM is to be interpreted as the owner of information, the second component as the set of readers, and the third component as the set of influencers. Note that RWFM is fully defined in terms of S, the set of subjects in the information system.

Note that the first component in the label is introduced only to facilitate limited discretionary flows (downgrades), and has no impact on the permissible information flows, or joins and meets. Therefore, we have abused notation in the above definition for simplicity, by uniformly blanking out the first component of the label.

Notation $\lambda : S \cup O \rightarrow S \times 2^S \times 2^S$ denotes a labelling function. $A_\lambda(e)$, $R_\lambda(e)$ and $W_\lambda(e)$ denote the first (owner/admin), second (readers), and third (writers) components of the security class assigned to an entity (subject or object) e. Further, the subscript λ is omitted when it is clear from the context. Note that $A(s) = s$ always.

Note that in RWFM information flows upwards in the lattice as readers decrease and writers increase.

Theorem 1 (Completeness). *RWFM is a complete model, w.r.t Denning's lattice model, for studying information flows in a system.*

The recasting procedure presented at the beginning of this section actually constructs such an equivalent RWFM policy for a given Denning's policy.

RWFM Semantics of Secure Information Flow RWFM provides a state transition semantics of secure information flow, which presents significant advantages and preserves useful invariants that aid in establishing that the system is secure or not misusing information. In the following, we present the RWFM semantics.

Let S denote the set of all the subjects in the system. RWFM follows a floating-label approach for subjects, with labels $(s, S, \{s\})$ and $(s, \{s\}, S)$ denoting the "**default label**" - the label below which a subject cannot write, and "**clearance**" - the label above which a subject cannot read, for a subject s respectively. Object labels are fixed and are provided by the desired policy at the time of their creation.

Definition 2 (State of Information System). *State of an information system is defined as the set of current subjects and objects in the system together with their current labels.*

Next, we describe the permissible state transitions of an information system, considering the primitive operations that cause information flows. The operations that are of interest are: (i) subject reads an object, (ii) subject writes an object, (iii) subject downgrades an object, and (iv) subject creates a new object. We believe that these operations are complete for studying information flows in a system. Note that we consider the set of subjects as fixed, and hence no operations for creation of new subjects.

For each of the above operations, we describe the conditions under which it is safe (causes only permissible information flows) and hence can be permitted. Note that when a subject s requests a new session, system assigns $(s, S, \{s\})$ as its current label.

READ Rule. *Subject s with label (s_1, R_1, W_1) requests read access to an object o with label (s_2, R_2, W_2).*
If $(s \in R_2)$ then
    change the label of s to $(s_1, R_1 \cap R_2, W_1 \cup W_2)$
    ALLOW
Else
    DENY

WRITE Rule. *Subject s with label (s_1, R_1, W_1) requests write access to an object o with label (s_2, R_2, W_2).*
If $(s \in W_2 \wedge R_1 \supseteq R_2 \wedge W_1 \subseteq W_2)$ then
    ALLOW
Else
    DENY

CREATE Rule. *Subject s labelled (s_1, R_1, W_1) requests to create an object o.*
Create a new object o, label it as (s_1, R_1, W_1) and add it to the set of objects O.

DOWNGRADE Rule. *Subject s with label (s_1, R_1, W_1) requests to downgrade an object o from its current label (s_2, R_2, W_2) to (s_3, R_3, W_3).*
If $(s \in R_2 \wedge s_1 = s_2 = s_3 \wedge R_1 = R_2 \wedge W_1 = W_2 = W_3 \wedge R_2 \subseteq R_3 \wedge (W_1 = \{s_1\} \vee (R_3 - R_2 \subseteq W_2)))$ then
    ALLOW
Else
    DENY

Intuitively, downgrading is allowed only by the owner at the same label as the information being downgraded, and (i) unrestricted addition of readers if he is the only influencer of information, or (ii) additional readers restricted to the set of stakeholders that contributed to the computation.

Given an initial set of objects on a lattice, the above transition system accurately computes the labels for the newly created information at various stages of the transaction/workflow.

The transition system above satisfies the following invariants that are handy to establish flow security:

1. subject and object labels float upwards only,
2. for a subject s, $A(s) = s$, $s \in R(s)$, and $s \in W(s)$,
3. the set of writers of information is always accurately maintained (exactly the set of subjects that influenced the information content), this plays a vital role in forensics and audit,
4. label of newly created objects precisely reflects the circumstances under which they are created,

5. downgrade rule is within the boundaries of the flows permissible under a given transaction, and
6. multiple sessions of the same subject are handled cleanly.

3 Formalizing and Analyzing Privacy Policies in RWFM

Information system is a collection of subjects and objects, where subjects are the users of the system and objects denote the information stored in the system.

Definition 3 (Data Usage/Privacy Policy). *Data usage or privacy policy of an information system (S, O), is defined as a function $\lambda : O \rightarrow S \times 2^S \times 2^S$ that assigns* `RWFM` *labels to the objects in the system.*

Given the intuitive nature of `RWFM` labels and their explicit use of readers and writers to encode the policy, we firmly believe that converting natural language descriptions of typical privacy policies - use terms like transfer, permit, prohibit, use, collect etc. - to `RWFM` labels should be much simpler than other languages and logics developed for this purpose.

Recall that, given two `RWFM` labels $L_1 = (s_1, R_1, W_1)$ and $L_2 = (s_2, R_2, W_2)$, we say that L_1 can-flow-to L_2 denoted by $L_1 \leqslant L_2$, if and only if $R_1 \supseteq R_2$ and $W_1 \subseteq W_2$ i.e., L_1 has more readers and fewer influencers compared to L_2.

Definition 4 (Policy Comparison ($\sqsubseteq$)). *Given an information system (S, O), and two policies λ_1 and λ_2, we say that:*

- *λ_1 is* **weaker** *than λ_2, written $\lambda_1 \sqsubseteq \lambda_2$, if and only if $\forall o \in O, \lambda_1(o) \leqslant \lambda_2(o)$,*
- *λ_1 is* **stronger** *than λ_2, written $\lambda_1 \sqsupseteq \lambda_2$, if and only if $\lambda_2 \sqsubseteq \lambda_1$, and*
- *λ_1 and λ_2 are* **incomparable***, if and only if $\lambda_1 \supseteq \lambda_2$ and $\lambda_2 \supseteq \lambda_1$.*

Intuitively, a weaker policy allows more readers for objects because of the superset ordering on the corresponding `RWFM` labels. Note that the relation $\sqsubseteq$ amongst policies is a partial-order.

Recall that, given two `RWFM` labels $L_1 = (s_1, R_1, W_1)$ and $L_2 = (s_2, R_2, W_2)$, we obtain label L on the combined information of the labels L_1 and L_2 denoted by $L_1 \oplus L_2$, as $(s, R_1 \cap R_2, W_1 \cup W_2)$.

Definition 5 (Policy Combination ($\boxplus$)). *Given an information system (S, O), and two policies λ_1 and λ_2, their combined policy λ denoted $\lambda_1 \boxplus \lambda_2$, is defined as follows: $\forall o \in O, \lambda(o) = \lambda_1(o) \oplus \lambda_2(o)$.*

Intuitively, the readers of an object in a combined policy are obtained by intersecting the readers of both the component policies. Note that the combined policy defined above is the weakest policy stronger than both the component policies.

Definition 6 (Policy Lattice). *The set of all possible policies of an information system (S, O), denoted by Λ, forms a lattice $(\Lambda, \sqsubseteq, \boxplus, \lambda_\perp)$, where $\lambda_\perp$ defined as $\forall o \in O, \lambda_\perp(o) = (s, S, \{\})$ denotes the weakest policy.*

3.1 Checking Policy Conformance in Federated Information Systems Through RWFM

Often two (or more) information systems fuse to form a federated system for advancing their business goals through sharing/exchanging data. A major concern in such systems is the security of data sharing i.e., compatibility of the data usage/privacy policies of the systems involved. In this section, we shall demonstrate how this can be effectively addressed through RWFM.

Definition 7 (Secure Information Sharing). *Consider information systems* $IS_1 = (S_1, O_1)$ *and* $IS_2 = (S_2, O_2)$ *with policies* λ_1 *and* λ_2 *respectively. We say that it is safe for* IS_1 *to share data* o_1 *with* IS_2 *if and only if,* $\lambda_1(o_1) \leqslant \lambda_2(o_1)$. *In this case, we also say that* IS_2 **conforms** *to* IS_1*'s policy on* o_1.

Intuitively, the above definition says that if the data receiver's policy allows fewer readers than the data owner's policy, then it is safe for the owner to share the data. Note that the asymmetry in the above definition captures the directionality of data movement.

Example 1. *Facebook's policy* (λ_1) *prohibits transfer of its user data* (o_1) *to advertising partners (ad) while it permits the use of this data by platform content providers like Zynga - in terms of* RWFM *labels this means* $ad \notin R(\lambda_1(o_1))$. *Zyngas policy* ($\lambda_2$) *permits transfer of user-id* (o_1) *to advertisers (ad) for preventing fraud - in terms of* RWFM *labels this means* $ad \in R(\lambda_2(o_1))$. *Facebook sharing user data with Zynga is insecure because* $\lambda_1(o_1) \not\leqslant \lambda_2(o_1)$, *because* $R(\lambda_1(o_1)) \not\supseteq R(\lambda_2(o_1))$, *because* $ad \notin R(\lambda_1(o_1))$ *but* $ad \in R(\lambda_2(o_1))$ *- a clear case of Zynga's non-conformance to Facebook's policy i.e., policy conflict.*

When multiple systems are combined to form a federated system, the set of subjects (objects) in the federated system is the union of the subjects (objects) of the component systems. While some objects in the federated system are exclusive to one component system, other objects - the objects that are agreed to be shared - may belong to multiple systems. However, note that there is exactly one component system that owns a shared object, and as far as the usage of that data is concerned the policy of the owner component must be the final word. Based on these intuitions, we now define a federated system of two component systems - can be generalized easily to $n > 2$ systems - as follows:

Definition 8 (Federated Information System). *The federated system of two information systems* $IS_1 = (S_1, O_1)$ *and* $IS_2 = (S_2, O_2)$ *with policies* λ_1 *and* λ_2, *is defined as a system* $FS = (S, O)$ *with policy* λ, *where* $S = S_1 \cup S_2$, $O = O_1 \cup O_2$, *and* $\forall o \in (O_{11} \cup O_{12}), \lambda(o) = \lambda_1(o)$, $\forall o \in (O_{22} \cup O_{21}), \lambda(o) = \lambda_2(o)$, *where*

$O_{11} = O_1 - O_2$ *Objects exclusive to* IS_1
$O_{22} = O_2 - O_1$ *Objects exclusive to* IS_2
$O_{12} \subseteq (O_1 \cap O_2)$ *Objects owned by* IS_1 *shared with* IS_2
$O_{21} \subseteq (O_1 \cap O_2)$ *Objects owned by* IS_2 *shared with* IS_1

Note that $(O_{12} \cap O_{21}) = \emptyset$, and $(O_{12} \cup O_{21}) = (O_1 \cap O_2)$.

Example 2. *Consider the scenario of Example 1. Assuming Facebook owns the user data o_1, in a federated system consisting of Facebook and Zynga, the policy on o_1 should be that of Facebook's i.e., $\lambda_1(o_1)$.*

Note how our definition of a federated system forces conformance to data owner's policy on its usage. As desired, the above definition reflects that in a federated system "owner of the data controls its usage", alternatively "in case of a conflict, the owners policy always wins".

Theorem 2. *Data sharing in a federated information system constructed per Definition 8 is secure.*

3.2 Enforcing Policy Conformance Through RWFM Dynamic Labelling

In the previous section, we have demonstrated how secure data sharing among multiple systems be achieved by combining them to form a federated system. Although conceptually simple, in practice it is not possible to achieve such a combined system due to a variety of reasons, and the systems have to operate from their individual domains. In this section, we shall show how even in this case `RWFM` dynamic labelling can achieve the desired security when sharing information that crosses domains.

Example 3. *Continuing the scenario in Examples 1 and 2, Facebook labels the user personal details (o_1) as $\lambda_1(o_1) = (F, \{F, Z, U\}, \{F, U\})$, where F, Z and U denote Facebook, Zynga and user respectively. If Z tries to read o_1 at a stage where Z's label is (Z, R_1, W_1), as per the READ rule of* `RWFM`*, it will be allowed because $Z \in R(\lambda_1(o_1))$, however Z's label will be changed to $(Z, R_1 \cap \{F, Z, U\}, W_1 \cup \{F, U\})$. As per the* `RWFM` *dynamic labelling rules, any object o that Z creates/writes after this step will have to satisfy $R(\lambda_2(o)) \subseteq (R_1 \cap \{F, Z, U\})$, which in turn implies $R(\lambda_2(o)) \subseteq \{F, Z, U\}$, which automatically enforces conformance with Facebook's policy.*

The example above demonstrates that dynamic labelling in `RWFM` i.e., semantics of operations `read`, `write`, and `create` in `RWFM` ensure that even if the receivers policy on data usage is in conflict with the owners policy, it is the owners policy that will be respected, alternatively, the receivers policy cannot override the owners policy. However, it is possible that the receivers policy is stronger than the owners policy, in which case the receivers policy already conforms to the owners policy and will be enforced.

The reason for such a result is the following: when an object o is fetched from a domain with policy $\lambda_1(o)$ into a domain with policy $\lambda_2(o)$, `RWFM` dynamic labelling rules ensure that the effective policy for o in the second domain would be $\lambda_1(o) \oplus \lambda_2(o)$, which is stronger than $\lambda_1(o)$ by definition and hence also conforms to $\lambda_1(o)$. The advantage of using `RWFM` is that this holds transitively i.e., even when the object crosses multiple domains.

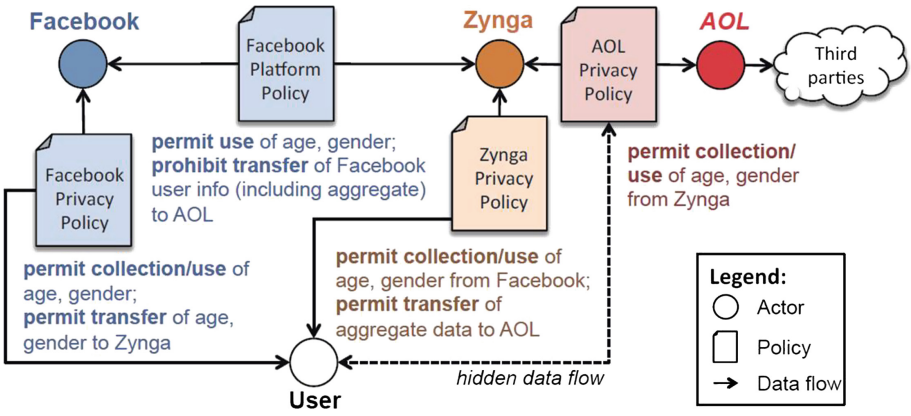

Fig. 1. Interaction of Facebook and Zynga privacy policies

Theorem 3. *Information systems implementing* `RWFM` *dynamic labelling can* **freely share data** *(labelled objects) amongst themselves - even through multiple hops - without the fear of compromising security, and worrying about the compatibility/conformance of the receivers policy.*

Illustrative End-to-End Example. Consider Facebook-Zynga policy in Fig. 1.

`RWFM` dynamic labelling for a typical usage scenario where a user logs in to his Facebook page, checks his messages, and plays a Zynga provided game, is depicted in Fig. 2. In the figure, states are represented by boxes and transitions are depicted by arrows labelled with operations responsible for the transition.

In the initial state, there are three subjects: F denoting Facebook, Z denoting Zynga, and U denoting the user, and three objects: CDb denoting the credential database of Facebook accounts, DZ denoting the default content provided by Zynga that is presented on the welcome page of the user, and UD denoting the contact details of the user. CDb is labelled $(F, \{F\}, \{F, U\})$, denoting that F is the owner, F is the only permissible reader, and F and U are the influencers - indeed password has to be provided by the user and hence his influence. DZ is labelled $(Z, \{F\}, \{Z\})$, denoting that Z is the owner and the only influencer, and F is the only permissible reader. UD is labelled $(F, \{F, U\}, \{F, U\})$, denoting that F owns it, F and U are the only influencers who are also the only permissible readers. Note that the label on UD can be automatically set based on the user's sharing preferences.

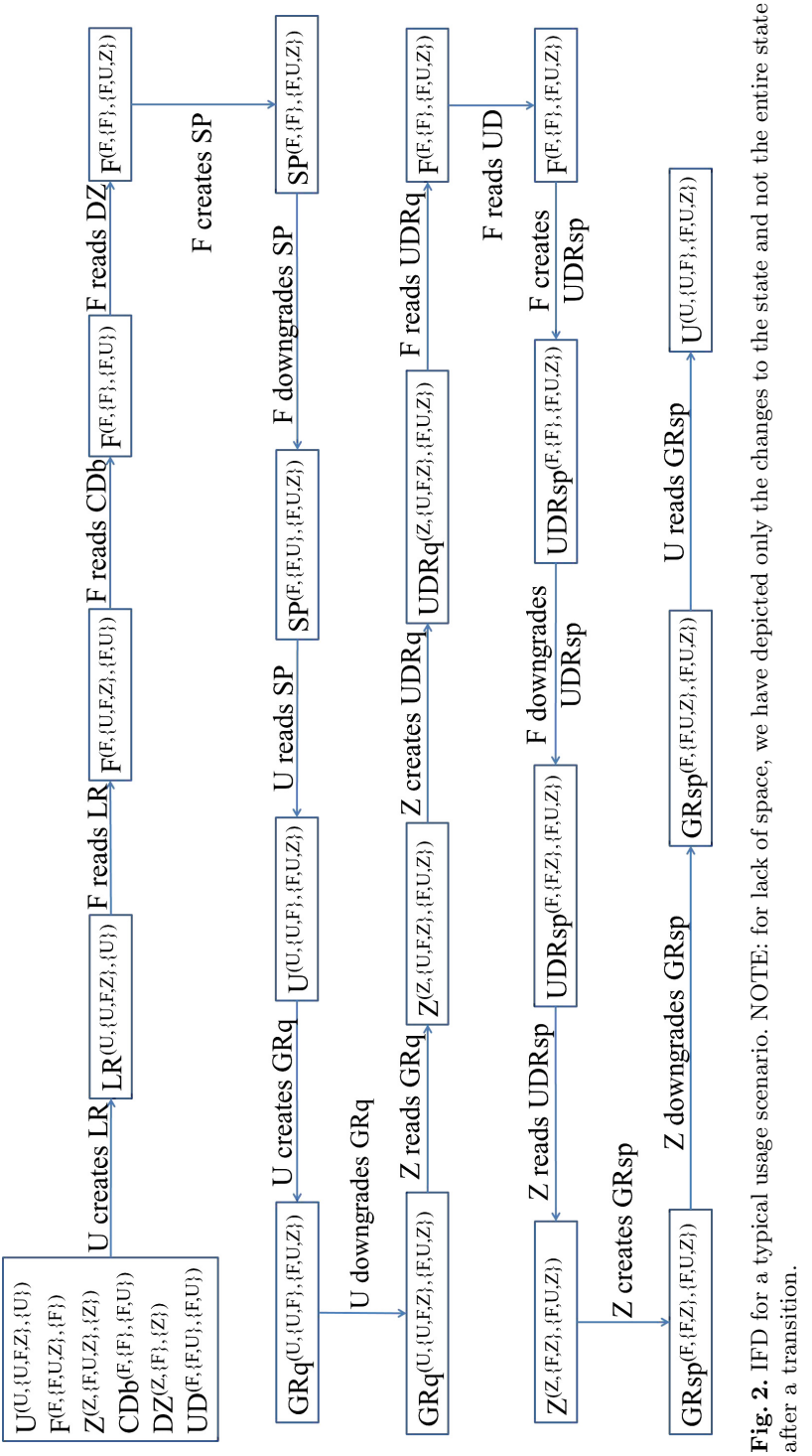

Fig. 2. IFD for a typical usage scenario. NOTE: for lack of space, we have depicted only the changes to the state and not the entire state after a transition.

Given the initial state and the operations performed by the user, observe how **RWFM** is able to automatically label all the subjects and new objects. Interpretation of the objects in the figure and are as follows:

LR - loginRequest
UD - userDetails
GRq - gameRequest
DZ - defaultZyngaContent
GRsp - gameResponse
SP - startupPage
CDb - credentialDatabase
UDRsp - userDetailsResponse
UDRq - userDetailsRequest

Note that, in the figure, we present only the changes to state, and not the full state, resulting from transitions. For example, second state in the system resulting from the transition labelled "U creates LR" is to be understood as adding the object LR, labelled $(U, \{U, F, Z\}, \{U\})$, to the system. A subject or object that is already a part of the system may also appear after a transition, subject due to read request and object due to downgrading, in which case this denotes label change only and not a new subject/object. This is the case, for example, at state 3 resulting from transition "F reads LR" and at state 7 resulting from transition "F downgrades SP".

From the figure, note that when F first creates the startup page (SP), F is its only permitted reader because before creation of SP, F had accessed the credential database which is sensitive and can only be accessed by F. However, because the startup page is prepared for the user, F downgrades it by adding U as a reader. In **RWFM**, this downgrading is allowed because SP has been created only upon the request from U and therefore influenced by him.

In Fig. 2, note that Facebook's response to Zynga's request for providing user details, denoted by UDRsp, is labelled (after downgrading) as $(F, \{F, Z\}, \{F, U, Z\})$. This guarantees that this information is not accessible by anyone other than Facebook and Zynga, thus, automatically forcing conformance on Zynga.

Further, note that the final labels of the three subjects are $F^{(F,\{F\},\{U,F,Z\})}$, $U^{(U,\{U,F\},\{U,F,Z\})}$, and $Z^{(Z,\{F,Z\},\{U,F,Z\})}$. From this we can derive that as far as this transaction is concerned, F dominates both U and Z in the hierarchy (because the readers of F is a subset of the readers of both U and Z). Intuitively, this says that U and Z could connect and interact only via F. This hierarchy inferred by our approach is depicted in Fig. 3.

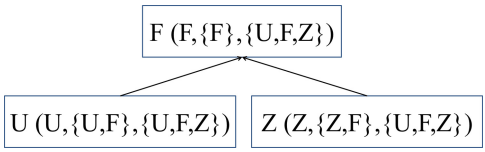

Fig. 3. Hierarchy of the subjects inferred by RWFM for the Facebook-Zynga example.

4 Application to Hybrid Cloud Security

ICT infrastructure requirements of organizations have grown several fold to meet the needs of its users, and has resulted in increasingly costly infrastructure proliferation. This increase in cost hindered organization's (particularly large heterogenous organizations) ability to modernise and fully exploit recent ICT developments.

Cloud computing is a way to access and use ICT services in a flexible and agile fashion, buying (hiring) only the services needed when they are needed. Cloud computing can be deployed through primarily four different models - private, public, hybrid and community. While the public cloud will offer substantial cost savings and increased flexibility for many ICT services and service users, data and privacy restrictions prevent some services from being hosted or provided through such means. The problem is particularly acute in the case of Government clouds. In these cases, a hybrid cloud service model can be used where the private cloud (managed in-house) provides the necessary security assurance to hold and process personal or restricted data.

Note that a hybrid cloud is a federated system, and that even in this case it is important to safeguard the data shared between the various clouds. Using the techniques sketched in the preceding section, we are working towards developing methods for data security and privacy from various perspectives in a hybrid cloud.

From a top-level view, the first step would be to secure the interactions between the clouds. This can be done at the level of a cloud manager, that is responsible for ensuring the authenticity of the stakeholders involved. Once this is achieved, then the next step would be to safeguard the privacy and security within a cloud when the computation is taking place. Towards achieving this, we are working on integrating an `RWFM` monitor with the map-reduce framework and the storage aspects including HDFS.

5 Comparison and Conclusions

In [1], authors encode data requirements from natural language privacy policies in Description Logic (DL), and analyse data flows within policies for detecting conflicts. Drawbacks of this approach compared to our approach are as follows: (i) DL is much harder to use compared to the intuitive nature of `RWFM` labels, (ii) no clear enforcement mechanism, and (iii) no remedies suggested for conflict resolution.

In [6], authors developed a system for automatic privacy compliance checking in big data systems and demonstrated its application to Bing. Drawbacks of this approach compared to our approach are as follows: (i) greater manual effort involved, (ii) works only for a centrally managed system, and (iii) does not control data propagation once released.

In this paper, we have demonstrated that: (i) `RWFM` labels provide a formalization for privacy policies in an intuitive way, (ii) `RWFM` labels provide an algorithm

for checking policy conformance and/or resolving policy conflicts, and (iii) the dynamic labelling approach of `RWFM` forces policy conformance in a distributed manner as data moves across multiple domains.

References

1. Breaux, T.D., Hibshi, H., Rao, A.: Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. Requirements Eng. **19**(3), 281–307 (2014)
2. Datta, A.: Privacy, audit and accountability. In: 22nd IFIP WCC 2012. http://www.wcc2012.org/pdfs/ADatta.pdf. Accessed 15 Aug 2016
3. Denning, D.E.: A lattice model of secure information flow. Comm. ACM **19**(5), 236–243 (1976)
4. Narendra Kumar, N.V., Shyamasundar, R.K.: Realizing purpose-based privacy policies succinctly via information-flow labels. In: 4th IEEE BDCloud, pp. 753–760. IEEE (2014)
5. Narendra Kumar, N.V., Shyamasundar, R.K.: POSTER: dynamic labelling for analyzing security protocols. In: 22nd ACM CCS, pp. 1665–1667 (2015)
6. Sen, S., Guha, S., Datta, A., Rajamani, S.K., Tsai, J.Y., Wing, J.M.: Bootstrapping privacy compliance in big data systems. In: IEEE SP, pp. 327–342 (2014)