Chapter 5
UPP+: A Flexible User Privacy Policy for Social Networking Services

Ramzi A. Haraty and Sally Massalkhy

Abstract Social networking services are having a major impact on people's daily lives. Ordinary users have taken these social networking facilities as basis for their businesses and for keeping track of their families and friends. In doing so, they add personal information, videos, pictures, and other data that is fundamentally unprotected due to the user's unawareness and the rigidity of the privacy policies of these facilities. Since users usually sign the privacy policy, granting their ownership of data to the site's owners, privacy concerns surface. In this paper, we present a privacy policy model—UPP+—for enhancing privacy and security for ordinary users. We use the Alloy language to formalize the model and the Alloy Analyzer to check for any inconsistencies.

5.1 Introduction

In the past few years, social network services have become major admirations in people's lives. Almost everyone who has access to the Internet has become addicted to certain social networking service sites such as Facebook, Google+, MySpace, or Twitter. In doing so, they add personal information, videos, and posts that are profoundly vulnerable due to the user's obliviousness and the stringency of the privacy policies of these facilities. Since users usually sign the privacy policy, granting their ownership of data to the site's owners, privacy concerns surface. Additional concerns arise when naïve users encounter [1]:

- Privacy policies that are hard to understand and assign particular policy settings that might conflict with each other.

R.A. Haraty (✉) • S. Massalkhy
Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102 2801, Lebanon
e-mail: rharaty@lau.edu.lb

R. Chbeir and B. Al Bouna (eds.), *Security and Privacy Preserving in Social Networks*, Lecture Notes in Social Networks, DOI 10.1007/978-3-7091-0894-9_5,

- Privacy policies that constantly change and keep on changing; thus, confusing the users.
- Privacy policies that are explained in an informal way and in an incomplete manner, which cannot provide consistent and complete account of the privacy.

The user's profile is usually the most important feature in a social network service. The owner of the profile is the one in control of the contents visible in the profile to others. Social network services often offer an access control panel that helps users control the privacy of their profiles by providing privacy policy levels whereby the user chooses a level and then categorizes her friends accordingly. This work presents an enhanced user privacy model, which we call UPP+. UPP+ is a flexible and easy to understand policy privacy for social networking services.

Before developing a policy, one needs to describe formally its components and the relationships between them by building a model. The model needs to be analyzed and checked to figure out possible bugs and problems. Thus, formalizing privacy security models helps designers building a consistent system that meets its requirements and respects the goals of discretion. This objective can be achieved through the Alloy language.

Alloy is a structural modeling language for software design. It is based on first order logic that makes use of variables, quantifiers, and predicates (Boolean functions) [2]. Alloy, developed by MIT (Daniel Jackson and his team), is mainly used to analyze object models, translates constraints to Boolean formulas (predicates), and then validates them using the Alloy Analyzer [3] by checking the code for conformance to a specification. Alloy is used in modeling policies, security models, and applications, including name servers, network configuration protocols, access control, telephony, scheduling, document structuring, and cryptography [4]. Alloy's approach demonstrates that it is possible to establish a framework for formally representing a program implementation and for formalizing the security rules defined by a security policy, enabling the verification of that program representation for adherence to the security policy [5, 6]. Additionally, it allows users to describe a system design and check that there is no misunderstanding before writing the code.

This remainder of this chapter is organized as follows: Sect. 5.2 provides related work. Section 5.3 presents the Alloy language, the Alloy Analyzer and their features. Section 5.4 presents the model descriptions and discusses the consistency proof and Sect. 5.5 concludes the work.

5.2 Related Work

There has been a plethora of work that deals with security and privacy policies. McLean [7] claimed that models are "used to describe any formal statement of a system's confidentiality, availability, or integrity requirements." Privacy models provide a detailed and precise means of formally describing privacy policies and proving their validity. Formalizing policy models provides system designers

with evidence that they are constructing a consistent system that will meet its specifications when implemented.

Fong et al. [8] proposed a privacy preservation model for social network sites like Facebook. In their paper, they analyzed and formalized the mechanism of the access control for Facebook social network. They imitated the Facebook's access control mechanism by taking into consideration its most important features which are its predicates.

Danezis [9] introduced a machine learning approach that was used to automatically find the privacy settings of users and give a readymade privacy policies package to the users; then this mechanism was evaluated. This approach is aimed to aid the end users when they want to restrict access from certain contacts. Their purpose is to infer user contexts, context assignment, and privacy policy per context.

Dania [10] introduced a formal model for social network privacy and used Facebook as her test case. Secure-UML was used as the formal language to the model. In [11], the author discusses the architecture, security policy, and protection mechanisms of four National Security Agency—certified systems. The author formally compares their techniques used for protecting data against users. In [12], the authors present a temporal multilevel secure data model. The model combines the characteristics of temporal data models and multilevel secure data models. The main focus of the model is mandatory access control, polyinstantiation, and secure transaction processing, while at the same time providing time support to record historical, present, and future data.

Hassan and Logrippo [13] proposed a method to detect inconsistencies of multiple security policies mixed together in one system and to report the inconsistencies at the time when the system is designed. The mixed models are checked for inconsistencies before real implementation. Inconsistency in a mixed model is due to the fact that the used models are incompatible and cannot be mixed. They demonstrated their method by mixing Bell–LaPadula with role-based access control (RBAC) [14] in addition to separation of concerns.

Shaffer in [15] described a security domain model (DM), designed for conducting static analysis of programs to identify illicit information flows, such as control dependency flaws and covert channel vulnerabilities. The model includes a formal definition for trusted subjects, which are granted privileges to perform system operations that require mandatory access control policy mechanisms imposed on normal subjects but are trusted not to degrade system security. The DM defines the concepts of program state, information flow, and security policy rules and specifies the behavior of a target program.

Misic and Misic in [16] addressed the networking and security architecture of healthcare information system. This system includes patient sensor networks, wireless local area networks belonging to organizational units at different levels of hierarchy, and the central medical database that holds the results of patient examinations and other relevant medical records. In order to protect the integrity and privacy of medical data, they targeted the Clinical Information System Security Policy and proposed the feasible enforcement mechanisms over the wireless hop.

The Clinical Information System Policy was recently formalized by Haraty and Naous [17].

The authors of [18, 19] presented a method to validate access control policy. They were mainly interested in higher level languages where access control rules can be specified in terms that are directly related to the roles and purposes of users. They discussed a paradigm more general than the RBAC in the sense that the RBAC can be expressed in them.

5.3 Formal Privacy Policy Model in Alloy

In this section, we overview the Alloy language and demonstrate how a model can be checked for consistency using Alloy and apply our method to our proposed UPP+ model.

5.3.1 The Alloy Language

To formalize the security models we use the Alloy language and its analyzer. Alloy is a lightweight modeling formalism using a first order predicate logic over the domain of relations. These relations are similar to relational algebra and calculus. It is a textual language developed at MIT. Alloy originates from Z. It is used for analyzing object models by checking for consistency of multiplicities and generating instances of models or a counterexample. Alloy Analyzer translates constraints to Boolean formulas and then applies SAT solvers.

5.3.2 Alloy Language Features

The following features present a subset of the full Alloy language that we used in formalizing our security models.

An Alloy model consists of one or more files, each containing a single module. A module consists of a header identifying the module, some imports and some paragraphs:

*module::=header import*paragraph**

A model can be contained entirely within one module. The paragraphs of module are signatures, facts, functions, predicates, assertions, run commands, and check commands.

Alloy uses the following multiplicity keywords: *lone*: zero or one; *one*: exactly one; *some*: one or more; *set*: zero or more. These keywords are used as quantifiers in quantified formulas, quantified expressions, in set declarations, in relation declarations, and in signature declarations.

A signature represents a set of atoms and is declared using the "sig" keyword—such as *sig A {}* to define a signature named *A*. The types of signatures are: *subset*, *top-level*, and *abstract*, and a signature with a *multiplicity* keyword:

- A top-level signature represents mutually disjoint sets that does not extend another signature: *sig A{}*
- A subset signature represents a set of elements that is a subset of the union of its parents: *one sig B extends A{}*
- An abstract signature represents only the elements that belong to one of the signatures that extend it: *abstract sig A{}*
- A signature with *multiplicity* keyword constrains the signature's set to have the number of elements specified with the keyword.

Facts, functions, and predicates are packages of constraints. A fact is a constraint that always holds. A predicate is a template for a constraint that can be instantiated in different contexts. A function is a template for an expression, and an assertion is a constraint that is intended to follow from the facts of a model. Examples of facts, predicates, and assertions are:

fact {no iden & parent}

pred access(state: State, next: state, u: User, r: Resource)
{next.accessed = state.accessed + u ->r}

*assert example*1 *{*
A.sens = SecretNT
B.sens = SecretT}

Run and *Check* commands are used to instruct the Alloy Analyzer to perform various analyses, a *run* command causes the analyzer to search for an instance that shows the consistency of a function or a predicate, whereas a check command causes it to search for a counterexample showing that an assertion does not hold:

check example
run UPP+Model

5.4 User Privacy Policy Plus (UPP+) Model

Aïmeur et al. [20], in their work, introduced the user privacy policy (UPP) model. UPP is a privacy model, which enables its users to control who can access their data in social networks. To understand UPP, the authors introduced a framework to the social networking sites that consists of user privacy concern, profile viewers, privacy levels, and tracking levels. In our work, we extend the UPP model to UPP+ and take into consideration part of UPP's privacy concern—the profile viewers and the privacy levels; however, the tracking levels will not be of importance and will not be implemented in our model.

The user privacy concern is split into three different categories. The first category is security, which is a major concern of social networking sites that deal with user's security risks such as identity theft and impersonation, hackers, phishing, and many others that may harm the user's data and information. The second category is reputation and credibility, which involves the reputation of the user—both online and in the real world, since bad reputation will lead to affecting the credibility of this user in the society or at work. The third category is profiling, which involves product companies building profiles on users from social network information found online without the user's knowledge in order to sell them products. In brief, tracking is following a user through a friend list or a name tag. There are three levels of tracking: strong tracking—a user is tracked on the social network; weak tracking—the user's name appears on the list of friends but not in tags; and no tracking—the user is not mentioned anywhere in his/her friends' profile.

Almost everyone now uses social networks; therefore, we have a variety of users. This variety of users leads to different concerns regarding the privacy. Aïmeur et al. [20] proposed four different types of privacy settings regarding the data of the user:

- *Healthy Data:* information, if shared, would return no harm to the user or even track him/her down such as nickname, music interest, and other similar data.
- *Harmless Data:* data, if shared, may contain data that helps in profiling for companies that market products. Such information are religion, gender, and interests.
- *Harmful Data:* information and pictures that belong to the user that, if shared, may lead to bad reputation and credibility.
- *Poisonous Data:* data that belongs to the user, if shared, may lead to security risks. Such information are user's home address, phone number, and other similar information.

After the process of data partitioning, friends partitioning is in order. Friends partitioning refers to who can access the different types of data. This is categorized into different groups of people depending on the relationship between the user and the friends s/he has on the social network site and according to the trust s/he has in his/her friends. In UPP, the people in the user's social network site are divided into four different groups:

- *Best Friends:* people who are considered the closest to the user such as parents or best friends.
- *Normal Friends:* people who are considered friends with the user but not necessarily close such as relatives and groups of friends.
- *Casual Friends:* people who are considered as somewhat strangers to the user yet known, like people the user met twice or friends of friends.
- *Visitors:* people who are strangers to the user. These people are not necessarily in the friends list.

UPP also introduces privacy that the user can choose from in order to have privacy setting to his/her social network page. The privacy levels are split into four

types. Each level has its own rules on each group of users. The rules are made in order to read the data sets:

- *No Privacy Rules*
  - Best friends can view all types of data set.
  - Normal friends can view all types of data set.
  - Casual friends can view all types of data set.
  - Visitors can view all types of data set.
- *Soft Privacy Rules*
  - Best friends can view all types of data set.
  - Normal friends can view Healthy data, Harmless data, Harmful Data but cannot view Poisonous data.
  - Casual friends can view Healthy data, Harmless data, Harmful Data but cannot view Poisonous data.
  - Visitors can view Healthy data and Harmless data but cannot view Harmful Data and Poisonous data.
- *Hard Privacy Rules*
  - Best friends can view all types of data set.
  - Normal friends can view Healthy data, Harmless data, Harmful Data but cannot view Poisonous data.
  - Casual friends can view Healthy data and Harmless data but cannot view Harmful Data and Poisonous data.
  - Visitors can view Healthy data but cannot view Harmless data, Harmful Data, and Poisonous data.
- *Full Privacy Rules*
  - Best friends can view all types of data set.
  - Normal friends can view Healthy data and Harmless data but cannot view Harmful Data and Poisonous data.
  - Casual friends can view Healthy data and Harmless data but cannot view Harmful Data and Poisonous data.
  - Visitors cannot view any type of data set.

  Table 5.1 summarizes the privacy settings, privacy levels, and users.

5.4.1 UPP+ Model Implementation

In order to implement UPP model, we will need to list the privacy data set, the privacy levels, the different types of users, and the constraints or the rules used in the model. This section will explain the implementation of this model. Table 5.2

Table 5.1 User privacy policy model

	Privacy settings				
Privacy levels	Healthy data	Harmless data	Harmful data	Poisonous data	Users
No Privacy	Yes	Yes	Yes	Yes	Best Friends
	Yes	Yes	Yes	Yes	Normal Friends
	Yes	Yes	Yes	Yes	Casual Friends
	Yes	Yes	Yes	Yes	Visitor
Soft Privacy	Yes	Yes	Yes	Yes	Best Friends
	Yes	Yes	Yes	No	Normal Friends
	Yes	Yes	Yes	No	Casual Friends
	Yes	Yes	No	No	Visitor
Hard Privacy	Yes	Yes	Yes	Yes	Best Friends
	Yes	Yes	Yes	No	Normal Friends
	Yes	Yes	No	No	Casual Friends
	Yes	No	No	No	Visitor
Full Privacy	Yes	Yes	Yes	Yes	Best Friends
	Yes	Yes	No	No	Normal Friends
	Yes	Yes	No	No	Casual Friends
	No	No	No	No	Visitor

Table 5.2 Privacy data set levels

Privacy data set levels	Description
PrivacyDS	Privacy Data Set
NoP	No Privacy Data Set
SoftP	Soft Privacy Data Set
HardP	Hard Privacy Data Set
FullP	Full Privacy Data Set

lists the privacy data set, while Table 5.3 lists the privacy data sets according to each level. Table 5.4 lists the user groups' data set and Table 5.5 lists the user groups according to each level.

The Privacy policies are split into four levels. In the No Privacy Level (NoP), a user in any category of No Privacy Users (Nusers), which are Best Friend (NBF), Normal friend (NNF), Casual friend (NCF), or Visitor (NV), has the right to read all four types of data, which are Healthy Data (NoPHealthyD), Harmless Data (NoPHarmlessD), Harmful Data (NoPHarmfulD), and Poisonous Data (NoPPoisonousD).

In the Soft Privacy Level (SoftP), a user (Nusers) from category Best friend (SBF) has the right to read all types of data; as is the case for Normal friend (SNF) and Casual friend (SCF)—they have the right to read Healthy Data (SoftHealthyD), Harmless Data (SoftHarmlessD), and Harmful Data (SoftHarmfulD) but cannot read Poisonous Data (SoftPoisonousD). A Visitor (SV) cannot read SoftHarmfulD and SoftPoisonousD.

Table 5.3 Privacy data set according to each level

No privacy data set	Description	Soft privacy data set	Description
NoPHealthyD	No Privacy Healthy Data	SoftPHealthyD	Soft Privacy Healthy Data
NoPHarmlessD	No Privacy Harmless Data	SoftPHarmlessD	Soft Privacy Harmless Data
NoPHarmfulD	No Privacy Harmful Data	SoftPHarmfulD	Soft Privacy Harmful Data
NoPPoisonousD	No Privacy Poisonous data	SoftPPoisonousD	Soft Privacy Poisonous Data
Hard privacy data set	**Description**	**Full privacy data set**	**Description**
HardPHealthyD	Hard Privacy Healthy Data	FullPHealthyD	Full Privacy Healthy Data
HardPHarmlessD	Hard Privacy Harmless Data	FullPHarmlessD	Full Privacy Harmless Data
HardPHarmfulD	Hard Privacy Harmful Data	FullPHarmfulD	Full Privacy Harmful Data
HardPPoisonousD	Hard Privacy Poisonous Data	FullPPoisonousD	Full Privacy Poisonous Data

Table 5.4 User group data set

Users group set	Description
Nusers	No Privacy User group
Susers	Soft Privacy User group
Husers	Hard Privacy User group
Fusers	Full Privacy User group

Table 5.5 Users groups according to each level

No privacy users group	Description	Soft privacy users group	Description
NBF	No Privacy Best Friend	SBF	Soft Privacy Best Friend
NNF	No Privacy Normal Friend	SNF	Soft Privacy Normal Friend
NCF	No Privacy Casual Friend	SCF	Soft Privacy Casual Friend
NV	No Privacy Visitor	SV	Soft Privacy Visitor
Hard privacy users group	**Description**	**Full privacy users group**	**Description**
HBF	Hard Privacy Best Friend	FBF	Full Privacy Best Friend
HNF	Hard Privacy Normal Friend	FNF	Full Privacy Normal Friend
HCF	Hard Privacy Casual Friend	FCF	Full Privacy Casual Friend
HV	Hard Privacy Visitor	FV	Full Privacy Visitor

In the Hard Privacy Level (HardP), a user (Husers) from category Best friend (HBF) can view all types of data. A Normal friend (HBF) cannot view Poisonous Data (HardPoisonousD). A Casual friend (HCF) and Visitor (HV) can view Healthy Data (HardHealthyD), Harmless Data (HardHarmlessD) but cannot view Harmful Data (HardHarmlessD) and HardPoisonousD.

In the Full Privacy Level (FullP), Best friend (FBS) can view all types of data, while Normal friend (FNF) and Casual friend (FCF) can view Healthy Data (FullHealthyD) and Harmless Data (FullHarmlessD) but cannot view Harmful Data (FullHarmfulD) and Poisonous Data (FullPoisonousD). As for Visitor (FV), s/he cannot view any type of data.

Table 5.6 UPP+ ownership

Ownership	Description
NO	No Privacy Owner
SO	Soft Privacy Owner
HO	Hard Privacy Owner
FO	Full Privacy Owner

Table 5.7 User privacy policy plus (UPP+) model

	Privacy settings				
Privacy levels	Healthy data	Harmless data	Harmful data	Poisonous data	Users
No Privacy	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Best Friends
	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Normal Friends
	Yes(r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Casual Friends
	Yes(r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Visitor
Soft Privacy	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Best Friends
	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	No (nr/nw/nc)	Normal Friends
	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	No (nr/nw/nc)	Casual Friends
	Yes (r/w/nc)	Yes (r/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	Visitor
Hard Privacy	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Best Friends
	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	No (nr/nw/nc)	Normal Friends
	Yes (r/w/nc)	Yes (r/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	Casual Friends
	Yes (r/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	Visitor
Full Privacy	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/w/nc)	Yes (r/nw/nc)	Best Friends
	Yes (r/w/nc)	Yes (r/w/nc)	No (nr/nw/nc)	No (nr/nw/nc)	Normal Friends
	Yes (r/nw/nc)	Yes (r/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	Casual Friends
	No (nr/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	No (nr/nw/nc)	Visitor

In our proposed UPP+ model, we added constraints to the UPP model to make it more plausible. The UPP model contains the Readby constraints, which shows the different levels of privacy and different types of users and suggests who can read the different types of data; however, nothing in the model suggested who can change these data or who can share them. Our contribution came by adding the changedby and the sharedby constraints. We maintained the readby constraints of the UPP model. Moreover, the type of users, the privacy levels, and the type of data being used in the UPP model were not changed.

Table 5.6 shows the ownership of the account in the UPP+ model, with NO standing for No Privacy Owner, SO standing for Soft Privacy Owner, HO standing for Hard Privacy Owner, and FO standing for Full Privacy Owner.

Our contribution is adding constraints to the model to see the consistencies when having the data changed or shared. "Sharedby" stands for the rules added to the model to grant or deny access to the users to share the data. "Changedby" stands for the rules added to the model to grant or deny access to the users to change and modify the data. Table 5.7 illustrates the constraints added. "r" indicates that a user

```
one sig NoPHealthyD,NoPHarmlessD,NoPHarmfulD,NoPPoisonousD extends NoP
                                {readby: some Nusers,sharedby: some Nusers, changedby: one Nusers}

one sig SoftPHealthyD,SoftPHarmlessD,SoftPHarmfulD,SoftPPoisonousD extends SoftP
                                {readby: some Susers,sharedby: some Susers, changedby: one Susers}

one sig HardPHealthyD,HardPHarmlessD,HardPHarmfulD,HardPPoisonousD extends HardP
                                {readby: some Husers,sharedby: some Husers, changedby: one Husers}

one sig FullPHealthyD,FullPHarmlessD,FullPHarmfulD,FullPPoisonousD extends FullP
                                {readby: some Fusers,sharedby: some Fusers, changedby: one Fusers}
```

Section 5.1 UPP+ declaration of privacy data sets in each level

```
//Declaration of Owner in Privacy Policy
one sig NO extends Nusers{}
one sig SO extends Susers{}
one sig HO extends Husers{}
one sig FO extends Fusers{}
```

Section 5.2 UPP+ owner's declaration set

```
//Declaration of owner's instances
one sig NO1 in NO{}
one sig SO1 in SO{}
one sig HO1 in HO{}
one sig FO1 in FO{}
```

Section 5.3 UPP+ owner's instance declaration set

can read the data, "nr" means a user cannot read the data, "w" indicates that a user can share the data, "nw" means a user cannot share the data, "c" indicates that the user can change the data, and "nc" means a user cannot change the data.

Section 5.1 shows the Privacy Data sets in each level as part of the Privacy levels (in the Alloy language). It displays the access rights given to the users.

Section 5.2 shows the declaration of the Ownership "Owner" in privacy policy which extends from the user's levels.

Section 5.3 shows the owner's instances that belong to each user level, which is similar to the users' instances.

Section 5.4 shows that at the No Privacy Level, NBF, NNF, NCF, and NV cannot share poisonous data, while all the users can share all other types of data. At the Soft Privacy Level, SBF, SNF, SCF, and SV cannot share poisonous data, SCF and SV cannot share harmful data, and SV cannot share harmless data, while the rest of the data is shared by the rest of the users.

Section 5.5 shows that at the Hard Privacy Level, HBF, HNF, HCF, and HV cannot share poisonous data; HNF, HCF, and HV cannot share harmful data; HCF and HV cannot share harmless data; and HV cannot share healthy data, while the rest of the users can share the rest of the data.

Section 5.4 UPP+ system sharedby constraints (part 1)

```
//Sharedby constraints on No Privacy data set

NoPPoisonousD.sharedby!=NBF
NoPPoisonousD.sharedby!=NNF
NoPPoisonousD.sharedby!=NCF
NoPPoisonousD.sharedby!=NV

//Sharedby constraints on Soft Privacy data set
SoftPPoisonousD.sharedby!=SBF
SoftPPoisonousD.sharedby!=SNF
SoftPPoisonousD.sharedby!=SCF
SoftPPoisonousD.sharedby!=SV

SoftPHarmfulD.sharedby!=SCF
SoftPHarmfulD.sharedby!=SV

SoftPHarmlessD.sharedby!=SV
```

Section 5.5 UPP+ system sharedby constraints (part 2)

```
//Sharedby constraints on Hard Privacy data set
HardPPoisonousD.sharedby!=HBF
HardPPoisonousD.sharedby!=HNF
HardPPoisonousD.sharedby!=HCF
HardPPoisonousD.sharedby!=HV

HardPHarmfulD.sharedby!=HNF
HardPHarmfulD.sharedby!=HCF
HardPHarmfulD.sharedby!=HV

HardPHarmlessD.sharedby!=HCF
HardPHarmlessD.sharedby!=HV

HardPHealthyD.sharedby!=HV
```

Section 5.6 UPP+ system sharedby constraints (part 3)

```
//Sharedby constraints on Full Privacy data set
FullPPoisonousD.sharedby!=FBF
FullPPoisonousD.sharedby!=FNF
FullPPoisonousD.sharedby!=FCF
FullPPoisonousD.sharedby!=FV

FullPHarmfulD.sharedby!=FNF
FullPHarmfulD.sharedby!=FCF
FullPHarmfulD.sharedby!=FV

FullPHarmlessD.sharedby!=FCF
FullPHarmlessD.sharedby!=FV

FullPHealthyD.sharedby!=FCF
FullPHealthyD.sharedby!=FV
```

Section 5.6 shows that at the Full Privacy Level, FBF, FNF, FCF, and FV cannot share poisonous data; FNF, FCF, and FV cannot share harmful data; FCF and FV cannot share harmless data; FCF and FV cannot share healthy data, while the rest of the users can share the rest of the data.

Section 5.7 shows that at the No Privacy Level and in Soft Privacy Level, all users of the model cannot change data, while the owner can change all types of data.

```
//Changedby constraints on No Privacy set
NoPPoisonousD.changedby!=NBF
NoPPoisonousD.changedby!=NNF
NoPPoisonousD.changedby!=NCF
NoPPoisonousD.changedby!=NV

NoPHarmfulD.changedby!=NBF
NoPHarmfulD.changedby!=NNF
NoPHarmfulD.changedby!=NCF
NoPHarmfulD.changedby!=NV

NoPHarmlessD.changedby!=NBF
NoPHarmlessD.changedby!=NNF
NoPHarmlessD.changedby!=NCF
NoPHarmlessD.changedby!=NV

NoPHealthyD.changedby!=NBF
NoPHealthyD.changedby!=NNF
NoPHealthyD.changedby!=NCF
NoPHealthyD.changedby!=NV
```

```
//Changedby constraints on Soft Privacy set
SoftPPoisonousD.changedby!=SBF
SoftPPoisonousD.changedby!=SNF
SoftPPoisonousD.changedby!=SCF
SoftPPoisonousD.changedby!=SV

SoftPHarmfulD.changedby!=SBF
SoftPHarmfulD.changedby!=SNF
SoftPHarmfulD.changedby!=SCF
SoftPHarmfulD.changedby!=SV

SoftPHarmlessD.changedby!=SBF
SoftPHarmlessD.changedby!=SNF
SoftPHarmlessD.changedby!=SCF
SoftPHarmlessD.changedby!=SV

SoftPHealthyD.changedby!=SBF
SoftPHealthyD.changedby!=SNF
SoftPHealthyD.changedby!=SCF
SoftPHealthyD.changedby!=SV
```

Section 5.7 UPP+ system changedby constraints (part 1)

```
//Changedby constraints on Hard Privacy set
HardPPoisonousD.changedby!=HBF
HardPPoisonousD.changedby!=HNF
HardPPoisonousD.changedby!=HCF
HardPPoisonousD.changedby!=HV

HardPHarmfulD.changedby!=HBF
HardPHarmfulD.changedby!=HNF
HardPHarmfulD.changedby!=HCF
HardPHarmfulD.changedby!=HV

HardPHarmlessD.changedby!=HBF
HardPHarmlessD.changedby!=HNF
HardPHarmlessD.changedby!=HCF
HardPHarmlessD.changedby!=HV

HardPHealthyD.changedby!=HBF
HardPHealthyD.changedby!=HNF
HardPHealthyD.changedby!=HCF
HardPHealthyD.changedby!=HV
```

```
//Changedby constraints on Full Privacy set
FullPPoisonousD.changedby!=FBF
FullPPoisonousD.changedby!=FNF
FullPPoisonousD.changedby!=FCF
FullPPoisonousD.changedby!=FV

FullPHarmfulD.changedby!=FBF
FullPHarmfulD.changedby!=FNF
FullPHarmfulD.changedby!=FCF
FullPHarmfulD.changedby!=FV

FullPHarmlessD.changedby!=FBF
FullPHarmlessD.changedby!=FNF
FullPHarmlessD.changedby!=FCF
FullPHarmlessD.changedby!=FV

FullPHealthyD.changedby!=FBF
FullPHealthyD.changedby!=FNF
FullPHealthyD.changedby!=FCF
FullPHealthyD.changedby!=FV
```

Section 5.8 UPP+ system changedby constraints (part 2)

Section 5.8 shows that at the Hard Privacy Level and in Full Privacy Level, all users of the model cannot change data, while the owner can change all types of data.

At this stage, we are ready to implement of the UPP+ model, using the Alloy language and its analyzer, in order to show its consistency. A Meta Model and instances are generated for the UPP+ model. Figure 5.1 depicts the Meta Model of UPP+. The figure shows that as in UPP model, PrivacyDS contains the four subsets: FUllP, HardP, NoP, and SoftP. Each of these levels contains the four types of data which are healthy, harmless, harmful, and poisonous, each of which extends

Fig. 5.1 The UPP+ meta model

```
Executing "Run test"
  Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
  542 vars. 265 primary vars. 463 clauses. 31ms.
  Instance found. Predicate is consistent. 47ms.
```

Fig. 5.2 UPP+ consistency output using Alloy Analyzer

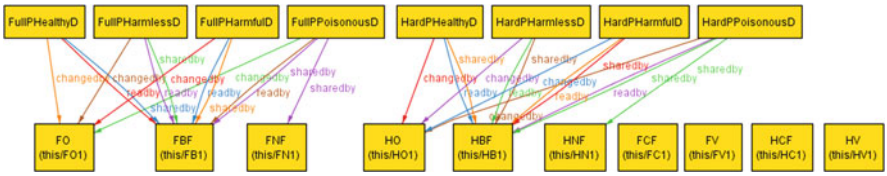

Fig. 5.3 UPP+ model instance 1 (part 1)

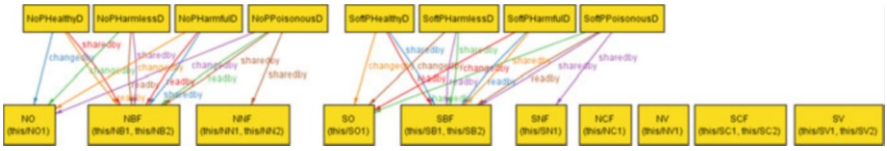

Fig. 5.4 UPP+ model instance 1 (part 2)

from the Privacy level. Since it is a Meta Model, it does not show the constraints of each user. The privacy data are read by, changed by, and shared by different types of users which are: Nusers, Susers, Husers, Fusers. Each type of the users extends to BF, NF, CF, V, and O.

After showing the Meta Model of UPP+, we need to test the model and show its constraints by running the predicate test. The result, depicted in Fig. 5.2, shows that instance is found and that the predicate is consistent. The time taken to check for consistency and to find an instance is 47 ms.

By clicking on Instance, the Alloy Analyzer will yield Figs. 5.3 and 5.4. More instances can be generated by clicking "next." Tables 5.8 and 5.9 show the instances using "changedby" and "sharedby."

Table 5.8 UPP+ model checking changedby consistencies

User type	Can change	Consistent
FO	FullPoisonousD	Yes
	FullHealthyD	Yes
	FullHarmlessD	Yes
	FullHarmfulD	Yes
HO	HardPoisonousD	Yes
	HardHealthyD	Yes
	HardHarmlessD	Yes
	HardHarmfulD	Yes
NO	NoPPoisonousD	Yes
	NoPHarmlessD	Yes
	NoPHarmfulD	Yes
	NoPHealthyD	Yes
SO	SoftPoisonousD	Yes
	SoftHealthyD	Yes
	SoftHarmlessD	Yes
	SoftHarmfulD	Yes

Table 5.9 UPP+ model checking sharedby consistencies

User type	Can share	Consistent
FBF	FullPoisonousD	Yes
	FullHealthyD	Yes
	FullHarmlessD	Yes
	FullHarmfulD	Yes
HBF	HardPoisonousD	Yes
	HardHealthyD	Yes
	HardHarmlessD	Yes
	HardHarmfulD	Yes
NBF	NoPPoisonousD	Yes
	NoPHarmlessD	Yes
	NoPHarmfulD	Yes
	NoPHealthyD	Yes
SBF	SoftPoisonousD	Yes
	SoftHealthyD	Yes
	SoftHarmlessD	Yes
	SoftHarmfulD	Yes

After showing that the system is consistent, we try different constraints that are wrong predicates, which should produce inconsistency. We ran a test example by stating that No Privacy visitor can change Harmless data as in Section 5.9 and that Full Privacy Best friend can share poisonous data, which proves the results of the predicate is inconsistent, as depicted in Fig. 5.5.

```
pred test()
{
NoPHarmlessD.changedby=NV
}
run test
```

```
pred test()
{
FullPPoisonousD.sharedby=FBF
}
run test
```

Section 5.9 UPP+ inconsistent predicates

```
Executing "Run test"
  Solver=sat4j Bitwidth=0 MaxSeq=0 SkolemDepth=1 Symmetry=20
  0 vars. 0 primary vars. 0 clauses. 16ms.
  No instance found. Predicate may be inconsistent. 0ms.
```

Fig. 5.5 UPP+ inconsistent output using Alloy Analyzer

5.5 Conclusion

In this work, we presented the UPP+ model, which carries significant enhancements over the UPP policy model. We used system examples based on the defined privacy model. We formalized the system according to the model and then checked its consistency and inconsistency. Since Alloy allows expressing systems as set of logical constraints in a logical language based on standard first order logic, we used it to define the system and its policy. When creating the model we specified the system users and data then Alloy compiles a Boolean matrix for the constraints, and we asked it to check if a model is valid, or if there are counterexamples. However, there exist many other privacy models for social networks that need to be formalized and analyzed to show their correctness. We plan to study these models in a more formal way to ascertain that they provide adequate privacy for users. We also believe that more work can be done to integrate multiple models in a mixed mode and formalize them to find potential interactions.

References

1. Danah, B.M., Ellison, N.: Social network sites: definition, history, and scholarship. J. Comput. Mediat. Commun. **13**(1), 210–230 (2007)
2. Jackson, D.: Alloy: a lightweight object modeling notation. Technical report 797. MIT Laboratory for Computer Science, Cambridge (2000)
3. Jackson, D., Schechter, I., Shlyakhter, I.: Alcoa: the alloy constraint analyzer. In: Proceedings of the International Conference on Software Engineering, Limerick, 2000
4. Wallace, C.: Using alloy in process modelling. Inf. Softw. Technol. J. **45**, 1031–1043 (2003). ISSN 0950-5849
5. Jackson, D.: Alloy 3.0 Reference Manual. Retrieved on April 18, 2013 from: http://alloy.mit.edu/reference-manual.pdf (2012)
6. Seater, R., Dennis, G.: Tutorial for Alloy Analyzer 4.0. Retrieved on April 18, 2013 from: http://alloy.mit.edu/tutorial4 (2012)
7. McLean, J.: In: Marciniak, J. (ed.) Encyclopedia of Software Engineering. Wiley, New York (1994)

8. Fong, P.W.L., Anwat, M., Zhao, Z.: A privacy preservation model for Facebook - style social network systems. In: Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS'09), Saint Malo. Lecture Notes in Computer Science, vol. 5789, pp. 303–320 (2009)
9. Danezis, G.: Inferring Privacy Policies for Social Networking Services. CCS Computer and Communications Security, pp. 5–10. ACM, New York (2009)
10. Dania, C.: Modeling social networking privacy. In: Doctoral Symposium of the International Symposium on Engineering Secure Software and Systems (ESSoS), pp. 49–54. CEUR, The Netherlands (2012)
11. Haraty, R.A.: C2 secure database management systems – a comparative study. In: Proceedings of the ACM Symposium on Applied Computing, San Antonio, 1999
12. Haraty, R.A., Bekaii, N.: Towards a temporal multilevel secure database. J. Comput. Sci. **2**(1) (2006). ISSN 1549-3636
13. Hassan, W., Logrippo, L.: Detecting inconsistencies of mixed secrecy models and business policies. Technical report. University of Ottawa, Ottawa (2009)
14. Ferraiolo, D.F., Kuhn, D.R.: Role-based access control. In: Proceedings of the 15th National Computer Security Conference, Baltimore, pp. 554–563, 1992
15. Shaffer, A., Auguston, M., Irvine, C., Levin, T.: A security domain model to assess software for exploitable covert channels. In: Proceedings of the ACM SIGPLAN Third Workshop on Programming Languages and Analysis for Security, pp. 45–56. ACM, Tucson (2008)
16. Misic, J., Misic, V.: Implementation of security policy for clinical information systems over wireless sensor networks. Ad Hoc Netw. J. **5**, 134–144 (2007). ISSN 1570-8705
17. Haraty, R.A., Naous, M.: Modeling and validating the clinical information systems policy using alloy. In: Proceedings of the Second International Conference on Health Information Science. Lecture Notes in Computer Science, pp. 1–17. Springer, London (2013)
18. Hassan, W., Logrippo, L.: Detecting inconsistencies of mixed secrecy models and business policies. Technical report. University of Ottawa, Ottawa (2009)
19. Haraty, R.A., Naous, M.: Role-based access control modeling and validation. In: Proceedings of the Fifth IEEE International Workshop on Performance Evaluation of Communications in Distributed Systems and Web based Service Architectures (PEDISWESA'2013), Split, 2013
20. Aïmeur, E., Gambs, S., Ho, A.: UPP: user privacy policy for social networking sites. In: Proceedings of the Fourth International Conference on Internet and Web Applications and Services, pp. 267–272, 2009