Privacy Aware Engineering: A Case Study

Xiaojun Ye and Zhiwu Zhu
Key Laboratory for Information System Security, Ministry of Education,
School of Software, Tsinghua University, Beijing 100084, China
Email: yexj@tsinghua.edu.cn

Yong Peng and Feng Xie
Evaluation Equipment R&D Lab, China Information Technology Security Evaluation Center,
Building A, No. 8 Yard of Shangdi Xilu, Haidian District, Beijing 100085, China
Email: pengy@itsec.gov.cn

Abstract—**Privacy is a complex social process that will persist in one form or another as a fundamental feature of software engineering. For successful privacy aware engineering, it is critical to guarantee the alignment and compliance among privacy artifacts emerging during software development process. In this paper, we propose a privacy compliance engineering flow in which we investigate the involved necessary privacy artifacts and discuss their alignment, refinement, and compliance verification. Within an exemplary case study, we identify the privacy artifacts introduced in the refinement process and analyze their compliance verification.**

Index Terms—**privacy, privacy engineering, privacy compliance, privacy enhancing technology**

I. Introduction

In the last years, privacy has become a critical issue in the development of IT systems, and current academic and industrial work have contributed many privacy engineering [2,19,20] and related privacy enhancing technologies (PETs)[12]. Unfortunately, it has always been difficult to protect personal private information in most privacy aware IT systems. For example, there is no obvious link between the privacy policy and the supporting process or component in software systems with current privacy enhancing technologies [19]. The observed main reason is that the current PETs focus mainly on the privacy policy implementation techniques alone and privacy engineering methodologies lack of effective approaches to align the process of privacy requirements elicitation, privacy policies definition, privacy modeling, and privacy enhancing technique implementation [19, 22].

Protecting user privacy is about complying with various public or organization regulations and user's desires when it comes to handling personal information. To implement the required privacy features with general software development process, approaches may include the following four phases: (1) understand what user privacy perceptions and privacy regulation requirements, (2) design the corresponding privacy goals or policies within the system framework, (3) transform them to formal privacy models and/or specify the equivalent privacy constraints, and finally (4) describe technological solution for assuring user privacy concerns during system implementation. Actually, this privacy engineering process is not always effective in practice as the compliance of privacy artifacts along the whole software development and evolution process cannot be guaranteed and verified [19,22].

To provide better privacy compliance engineering solution, it is indispensable to integrate the privacy engineering methodologies with the privacy enhancing technologies compliance checking. In this paper, we are towards finding the issues that degrade the above process feasibility and proposing the corresponding methodological solution to improve the privacy artifacts compliance validity along the privacy aware software development process.

We define *privacy aware engineering* as a systematic integration of privacy artifacts introduced during software development with consistent privacy semantic checking and privacy features coverage checking. It is an engineering methodology that aligns the sequentially produced and evolved privacy artifacts in a compatible, compliable, and verifiable way. In other words, we must consider the impact of every component or process on individuals' privacy in software development and do this throughout the software lifecycle, thus ensuring that appropriate privacy controls are implemented and maintained in IT- systems.

To specify such a methodology, we propose the *privacy engineering flow*, in which we discuss the involved privacy artifacts from the earliest stages of the system business goal, through privacy requirements gathering, to delivery, testing, operations, and out to the final decommission of the IT-system. It illustrates the elaborate process from privacy goals definition based on experienced privacy principles to privacy constraints specification, through privacy artifacts refinement, evolution, and iteration. For better understanding this methodology, we give an exemplary case study, in which we investigate privacy artifacts and analyze their compliance verification during the privacy software engineering process.

The remainder of the paper is organized as follows. Section 2 gives a brief definition and an interpretation for several privacy-related artifacts involved in the privacy-aware software development process. Section 3 defines the privacy aware engineering notion, including the privacy engineering flow, privacy artifacts refinement process and privacy compliance verification. Section 4 gives a case study, including all privacy artifact types involved in privacy engineering and their compliance relations. Section 5 discusses the related work in privacy policies and engineering. Finally, Section 6 gives a short conclusion for the paper.

II. Preliminary Concept

In this section, we discuss the concepts necessary to understand and express privacy aware engineering process. The term *privacy* means many things in different contexts. In this paper, we consider *data privacy* which refers to the evolving relationship between technology and the legal right to, or public expectation of privacy in the collection and sharing of sensitive data about ones-self in the electronic environment. So the notion privacy is regarded as the claim of individuals, groups and institutions to determine for themselves when, how and to what extent information about them is communicated to others [1, 4]. One issue in the privacy enhancing technologies is how to ensure compliance with various privacy principles and privacy goals specified during the software requirement analysis phase [2,3,7].

Privacy principle is a general experienced observation of privacy protection from privacy laws/regulations in generic practice. For example, in 1980 the OCED published eight principles/guidelines on the protection of privacy and trans-border flow of personal data [1]. These principles emphasize the need to minimize the collection and use of personal data, to inform individuals about data collection, and to adequately maintain and protect collected data. *Privacy goal* is taken as an expectation of the privacy-enhancing work according to privacy principles and user’s desires during system development, i.e., state of affairs that need to be attained [5]. The elicitation of the privacy goals, which are relevant to the specific organization, is the first step towards privacy engineering. It is a presystem conceptual representation of the privacy features which will be supported by the software. Antón et al. have structured the privacy policy domain with goal taxonomies and classified them as either privacy protection goals or privacy vulnerability goals[6]. The former are related to privacy principles and data subject rights such as notice/awareness, choice/consent, access/participation, integrity/security, and enforcement/redress, the latter concern privacy threats and are classified in seven categories, namely monitoring, aggregation, storage, transfer, collection, personalization, and contact. Kalloniatis et al. review current security and privacy requirements research, and conclude privacy goals mainly as the following eight privacy requirements: *identification, authentication, authorization, data protection, anonymity, pseudonymity, unlinkability and unobservability* [5,6,17,19]. In fact, a number of security goals influence privacy goals. This is the reason for including a number of security related requirements in the privacy goals set. By addressing these requirements one aims to minimize or eliminate the collection of user identifiable data.

IT-systems must inform data provider/subject about the use of their personal data by means of privacy policy. So *privacy policy* is another abstract description of the privacy goals of a specific system. Similar to goals decomposition, privacy policies may be decomposed in simpler policies or may support/conflict the achievement of other policies. As such, a privacy policy can be *high-level* or *lower-level*. *High-level privacy policies* directly reflect applicable laws, regulations, or agreements which are related to the system-specific *privacy principles*/*goals*, and ideally should be applicable in software systems. While *lower-level privacy policies* concern more on the software technologies level privacy requirements for the system implementation, they contain both references to what are stipulated by high-level policies and what will actually be implemented by privacy enhancing technologies.

Privacy requirement, as a special type of privacy goals which constraint the causal transformation of privacy policies into models and mechanisms [7, 9, 19], is also a system-specific manifestation of privacy policies in an application, such as: *Users submit manuscripts as “Author”, which can be downloaded by “Reviewers” and “Administrators”*. Defining privacy requirements and bringing them into alignment with the software development process are complex activities. These activities require one to understand what are organization goals, the structure of the organization and its environmental context. Identifying privacy requirements may be guided by the different privacy goals introduced above. Note that privacy requirements are tightly related with security requirements which are defined as constraints on non-functional requirements in aspect of security and vulnerability [6, 17]. Each privacy requirement is elicited and extracted based on high-level privacy policies [4, 5]. So lower-level privacy policies can be subsequently defined under the guideline of high-level privacy requirements, as an instance: *Allow users to register as an ”Author" by providing email address.*

In the above, we have discussed diverse privacy artifacts for the specification of user privacy concerns. Organization should adopt measures to enforce these privacy features by lower-level privacy policies. They are used to specify the privacy features that should be supported by privacy aware processes implemented in backend systems or frontend applications respectively [9, 15, 16]. A privacy aware process indicates that the specific privacy-related activities implementation technique needs to be introduced in order to ensure that the process is privacy compliant. Naturally, the choice of the appropriate implementation technique depends on the privacy requirement(s) under consideration. Kalloniatis et al. have introduced seven privacy-process patterns corresponding to the eight basic privacy goals, i.e., generalized process models which include activities and flows connecting them, presenting how a business should be run in a specific domain [19].

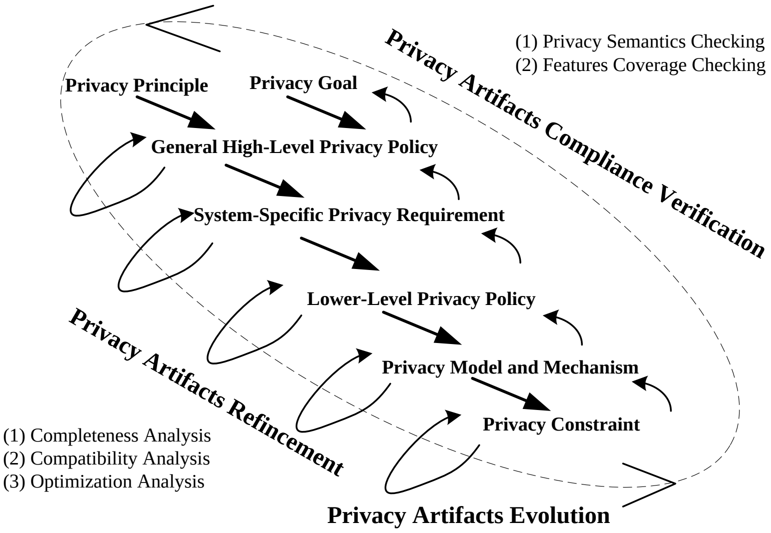
Lower-level privacy policies can be both on the database level and on the application level. Most lower-level data privacy policies are on database level since privacy preserving, as well as security guaranteeing, is *data-centric* and *data-specific*. While in database systems, each lower-level privacy policy can be seen as a database-understandable abstraction of *privacy models* and *privacy enforcement mechanisms* [9]. *Privacy model* is a well-organized composition of some privacy mechanisms for completely reaching a generalized privacy requirement, where a *privacy enforcement mechanism* is a lower-level access control functional description of privacy policies, including a list of privacy rules connecting the outside users and the inside data properties, and a *privacy rule* is a concrete control (restriction) description implemented by the database system to perform the specific privacy protection mechanism. Privacy rules can be classified into three types: *conditions*, *constraints* and *obligations*, which respectively, indicate what restrictions should be satisfied before, during, and after data access [1, 7, 10].

Privacy model can be seen as a technique view of one or more lower-level privacy policies [16]. Each model defines a framework of a privacy protection technique for covering the relative privacy policies. Privacy models can be easily proved to be equivalent in expression capability to the lower-level privacy policies. Privacy model can be defined and improved based on existed security models and mechanisms [8]. As we know, commercial database systems are designed as generic as possible for the most applications. Privacy models and security models implemented in the databases are often stable and generalized for the widest usage population [1]. So a new privacy model can inherit a part of existed security/privacy modeling work while guarantee the special components compatible with the existed privacy components, mechanisms, even models, for the system-specific privacy protection implementation [12].

III. PRIVACY AWARE ENGINEERING

In the above section, we give various privacy artifacts involved in privacy preserving software development process. To support the privacy aware systems development, researcher propose privacy requirements engineering or privacy engineering in order to organize and collaborate the above all privacy artifacts in a compatible and compliable way [3, 15, 20, 22]. Kenny et al. have firstly defined privacy engineering "as a systematic effort to embed privacy relevant legal primitives into technical and governance design"[20]. In this paper, we define

Privacy engineering is a methodology for incorporating privacy requirements into system design and implementation process, which gives a compatible, compliable, and verifiable privacy artifact definition, refinement and evolution approach from privacy goals and requirements, through software privacy modeling and constraints to various privacy-preserving system components and process implementation.



Privacy Principle
Privacy Goal
General High-Level Privacy Policy
System-Specific Privacy Requirement
Lower-Level Privacy Policy
Privacy Model and Mechanism
Privacy Constraint
Privacy Artifacts Compliance Verification
(1) Privacy Semantics Checking
(2) Features Coverage Checking
Privacy Artifacts Refincement
(1) Completeness Analysis
(2) Compatibility Analysis
(3) Optimization Analysis
Privacy Artifacts Evolution

Figure 1. Privacy artifacts evolution in the privacy engineering flow

The purpose of privacy engineering is to give due consideration to user privacy needs in the full lifecycle of system development process - in other words, to consider the impact of a process or a component on individuals' privacy and to do this throughout the system lifecycle, thus ensuring that appropriate control mechanisms are implemented and maintained. As an engineering process, it must answer the following three questions.

- How to elicit, define and refine privacy requirements?

- How to evolve the privacy requirements into the system-independent privacy models?
- How to verify and use the system-specific privacy models implemented in systems?

So we discuss privacy engineering through the *privacy engineering flow*, which unifies the all privacy artifacts as the way to guarantee a “better" privacy practice.

A. *Privacy Engineering Flow*

We define the *privacy engineering flow* as *the process of handling privacy artifacts and their relations during privacy software development*. This lifetime approach, illustrated in Fig.1, ensures that privacy controls are stronger, simpler and therefore cheaper to implement, harder to by-pass, and fully embedded in the system as part of its core functionality. Firstly, software analysts might collect privacy regulations and user desires in order to derive the privacy goals that give a preliminary understanding for the privacy features which should be provided in applications. Based on the privacy goals and the existed privacy practices (if any), privacy engineers define the high-level privacy policies for a specific system with privacy policy languages (such as P3P/EPAL/XACML), and subsequently, elicit the derived privacy requirements (low-level privacy policies) for the system developing, which should be compatible with other kinds of requirements, especially security requirements [4]. After privacy requirements refinement, privacy engineers can further extract additional lower-level privacy policies that specify the system privacy enforcement in a more precise way [19]. Mostly, these policies are manifested on data constraints on database level, since the protected private data are collected and stored in databases and served access control mechanism for outside user accesses [1, 8, 9].

From the perspective of database systems, lower-level privacy policies should be supported through revising and extending the existed security (or privacy) models and enforcement mechanisms. Privacy engineers need verify these produced privacy artifacts compliance as reversing the above direction and iterating the privacy evolution process till the privacy features for the application are fully captured by privacy requirements, supported by lower-level privacy policies, specified by privacy models and mechanisms, and implemented by privacy constraints in database systems.

B. *Privacy Artifacts Refinement Process*

From the senior-level to junior-level privacy artifacts transformation, senior-level or ‘strategic’ artifact may be decomposed in more simple artifacts, in other words, each ‘strategic’ privacy artifact may include a set of junior-artifacts. The aim of privacy artifacts refinement is to interpret the general privacy features with respect to the specific application context into consideration. Introduction of new privacy artifacts may lead to the emerging of new processes/components while improvement/adaptation of privacy artifacts may lead to the adaptation of associated processes accordingly. Repeating this process for every privacy artifact and its associated user desire privacy goals leads to the identification of alternative ways for resolving privacy requirements.

Along the privacy engineering flow, a privacy artifact refinement process may include the following tasks. (1) *Completeness analysis*. It is to make sure the current level privacy artifacts comply with the senior in aspect of the privacy features covered; (2) *Compatibility analysis*. It is to make sure the subartifacts in the current level compatible with each other, i.e. no semantic conflicts among them; (3) *Optimization analysis*. It is to make the final artifacts optimized in privacy engineering, i.e. the produced subartifacts are unambiguous and concise in privacy semantics expression.

Privacy artifact refinement refers to the process of completing its privacy semantics, making all junior-artifacts compatible, and refining into optimized forms.

To use the above privacy engineering flow in real practice, we need firstly understand two properties of privacy artifacts: *compatibility* and *compliance*.

The property of *compatibility* indicates that junior-artifacts are mutual compatible, i.e. they share no conflicts between the senior-level artifacts and its sub-artifacts. Such conflicts should be made explicit and resolved through negotiation among the various artifacts during privacy artifact refinement process. As such, these artifacts are produced along the privacy engineering process path. The property of *compliance* indicates junior-level artifacts can exactly express the privacy semantics of their senior-level artifact.

In detail, the compatibility analysis is to check the conflict situations among junior-artifacts. *Artifacts conflicts* refer to the artifacts that have conflict privacy semantics, which may incur conflict privacy features in the system developing. Privacy engineers might adopt some strategies to solve these conflict artifacts, such as removing the lower-priority artifacts, revising the latest-revised artifacts, etc. During this process, initial artifacts may get rephrased, some of them may be rejected and additional artifacts may be identified. The compliance verification is to check consistent privacy semantics and privacy features coverage among senior-level and junior-level artifacts.

C. *Privacy Artifacts Compliance Verification*

Privacy compliance is frequently made during artifacts refinement process [2]. Under this view, conforming to a privacy specification or policy, regulation or law that has been clearly defined as privacy artifacts during the system development should be included in privacy engineering [20]. How the privacy artifacts compliance is satisfied depends on the trust assumptions made mainly by the completeness analysis during the refinement process [15].

A set of junior privacy artifacts complies with the senior privacy artifact if and only if it captures all the privacy features of the senior, i.e. it is a more concrete and correct version of the senior in privacy semantics towards being more precisely understandable by privacy engineers.

Along the privacy engineering flow, each junior privacy artifact is to transmit and concretize the privacy semantics of the senior step by step, till the privacy model and/or constraints in database systems, which can be seen as the final status of privacy features in applications that can be directly implemented on database level.

The compliance verification mainly includes two aspects: *consistent privacy semantics checking* and *privacy features coverage checking*. The former is to verify if the privacy semantics by the current artifacts is consistent with that of the senior. The latter is to verify if the privacy features captured by the current artifacts equal to those by the senior. Privacy coverage checking can be used to verify if a set of junior-level sub-artifacts covers all possible privacy feature of senior-level privacy artifact.

From the semantics perspective, we can distinct characteristics of privacy artifacts, such as what actions are allowed, required or prohibited on privacy data. We can format privacy artifacts meta-data with the 5-tuple <*Subject, Action, Object, Condition, Mode*>, where *Action* is an operation by the subject (i.e. user) on the object (e.g., data), such as *use, access, read*, etc. *Mode* is *Allow* or *Deny*, indicating if the action is allowed when *Condition* is satisfied. Similar to the access control rule (such as in XACML), we can use a rule-based approach to describe the meaning of privacy artifacts and their relations as a policy set, and establish a basis of reasoning about these privacy artifacts [10].

The compliance verification assures the trust relationship involving privacy artifacts. Privacy artifacts in a relationship trust each other to have or not to have certain properties (the so-called privacy features). If junior-level privacy artifacts satisfy these properties of the high-level one, then they are trustworthy. Haley et al. have proposed a similar trust assumptions in security requirements analysis[15].

IV. A Case Study

Suppose we are required to develop a privacy aware website application similar to Conference Management Tool (CMT) [1]. Usually privacy principles/goals are individual statements, expressed in a form of natural language, specifying the behaviors and constraints of a proposed application system. We can firstly specify some privacy principles (PP) such as listed in Table I. Principle PP_1 is a security requirement, while principle PP_2 is derived from regulations US Privacy Act of 1974 and OECD privacy guidelines or from industry standard like P3P policies proposed by W3C organization.

Privacy principle/goal identification is the preparation work for system-specific requirements analysis, including domain engineering (user analysis and organization modeling), existed privacy principle collection and general privacy goals definition based on the former work. Based on these principles/goals, we can elicit, define and refine the system-specific privacy requirements based on the before-requirement phase. It includes the upper level privacy policy definition as the abstraction forms of privacy requirements and the lower level privacy policy definition following requirements specification based on privacy engineering flow.

TABLE I. Privacy Principles (PP)

PP_1	**Security dependence**: privacy protection should base on data security techniques;
PP_2	**Preference personalization**: privacy policies on private information should be personalized by information donors.

A. Privacy Artifacts Evolution

From internet privacy requirements, this website application should have the ability to control what information the register reveals about oneself over the Internet, and to control their private information collecting by most web applications and third parties. As such, we can propose another two privacy goals (PG) from the web-applications characteristics practices derived from OCED regulation guidelines and user's desires, such as listed in Table II.

TABLE II. Privacy Goals (PG)

PG_1	Avoid personal information violation during the retention;
PG_2	Use personal information to do privacy-related tasks.

Considering the privacy principles and goals, we can extract the high-level privacy policies (HLPP) for this specific Conference Management Tool. In the following, we just consider the refined privacy artifacts that are consistent with the two privacy goals specified in Table II. Two corresponding high-level privacy policies are list in Table III.

TABLE III. HighLevel Privacy Policies (HLPP)

$HLPP_1$	Protect personal information and manuscripts collected in database against illegal usage and disclosure during the retention;
$HLPP_2$	Use personal information for user identification and contraction.

Privacy requirement analysis typically starts with a detailed understanding of the relevant processes/components as well as privacy artifacts surrounding these processes/components. In the context of privacy engineering, we therefore need to understand what privacy regulations, user privacy perceptions and expectations exist, and how they might be compromised by IT system processes/functions (data collection, storage, and processing).

Based on the above high-level privacy policies and the privacy statement in CMT, we can elicit the privacy requirements (PR) during the requirement analysis phase, in aspect of IT system processes such as personal information collection, use, and security, see Table IV.

TABLE IV. Privacy Requirements (PR)

[1] Http://msrcmt.research.microsoft.com/cmt/

PR_{11}	User submits papers as “Author", which can be downloaded by “Reviewers" and “Administrators";
PR_{12}	The website shall collect personal identity or contract information for identifying and contracting users further;
PR_{21}	Personal information should be used with appropriate purposes;
PR_{22}	User may review and edit the personal information during registration;
PR_{31}	Personal information should not be shared outside the website;
PR_{32}	The website shall use security technologies to help protect personal information collected from unauthorized access, use, or disclosure.

During the system design phase, we can define the concrete lower-level privacy policies (LLPP) on application or database level, based on the above privacy requirements, which are used as the set of privacy features that will be implemented in applications for developing the website system, as in Table V.

TALBE V. LOWERLEVEL PRIVACY POLICIES (LLPP)

$LLPP_{11}$	Allow users to register as an “Author" by providing email address;
$LLPP_{12}$	Allow “Authors" to edit personal information and privacy preference;
$LLPP_{21}$	“Administrator" assigns users “Reviewers" for reviewing papers, which can be reviewed only by “Administrator" and “Reviewers";
$LLPP_{22}$	Only “Administrator" can access personal information of “Authors";
$LLPP_{23}$	Personal information should be used with appropriate purposes, as “identity", “sensitive", “review", “statistic", and “publicable";
$LLPP_{31}$	Use secure transmission technique to avoid transmission disclosure;
$LLPP_{32}$	Define personal information retention based on each conference case.

Under the guideline of the above lower-level privacy policies, we can choose the required privacy models and mechanisms during the database system design phase. We take purpose-based access control (PBAC) [8] as the privacy model to capture the features implied in LLPP and PR, we can derive the corresponding database privacy models/mechanisms in Table VI. In PBAC, purpose is specfied on users/roles and objects. For convenience, we use $r_i \leq r_j$ to indicate that role r_j *inherits* role r_i, and $p_i \leq p_j$ to indicate that purpose p_j is more *generalized* than purpose p_i. Further, we take $u_i \rightarrow op\ ob_j$ and $u_i \leftarrow op\ ob_j$ to indicate user u_i submits an access request (with operation *op*) to object ob_j and ob_j can be accessed by u_i by *op*, respectively.

TABLE VI. PRIVACY MECHANISMS AND MODELS(PMO, PME)

PMo_1	Adopt purpose-based access control (PBAC) model.
PM_{e1}	Adopt role-based access control (RBAC) model as the security base;
PM_{e2}	Adopt purpose mechanism to personalize privacy preferences;.
PM_{e3}	Adopt the Secure Socket Layer (SSL) protocol for sensitive personal information transmission;

To implement the above privacy mechanisms and models, we first need to define some basic privacy structures (meta-data) during the detail database design phase, which might be created or predefined in database management system, as in Table VII.

TABLE VII. PRIVACY STRUCTURES

Table	***Attribute***	***Denotatio n***
Role-Role	{RoleName, RoleName}	$r_i \leq r_j$
User-Role	{UserName, RoleName}	$ui \leftharpoonup rj$
Role-Privilege	{RoleName, Operation, ObjectName}	$<r_i,op_j,ob_k >$
Purpose-Purpose	{PurposeName, PurposeName}	$p_i \preceq p_j$
User/Role-Purpose	{UserName/RoleName,PurposeName}	u_i(or r_j) $\leftharpoondown p_k$
Object-Purpose	{ObjectName, PurposeName}	$ob_i \leftharpoondown p_j$

From Table VII, we can find that the first three meta data are used to describe the basic security mechanisms, the other three elements are:

- Purpose-purpose relations, where, there is a generalization/specialization relation between two hierarchical purposes. But beside it, there are some other important relations such as AND or OR (de)composition relations, static/dynamic mutual exclusion relations. AND/OR-(de)composition indicates a purpose is satisfied if only if all (for AND) or exist one (for OR) sub-purposes are satisfied;
- Purpose-role (or user+role) relations, where, purposes can be attached to a role or a role+user pair, indicating the role or the user who activates the role can legally use the purpose to access some private data;
- Object-purpose (purpose-privilege) relations. We define a privilege as an object and purpose on the action of object pair. So an object-purpose relation indicating the purpose-privilege is attached to the data while the operation can be allowably performed on the data by comparing with the access purpose in purpose-role relations.

In conclusion, privacy aware engineering procecss requires the privacy-aware database system to offer the elegant privacy components/mechanisms that can be generally used for implementing privacy requirements derived from privacy engineering in database systems [1].

With these privacy structures, we can derive the elegant privacy constraints to capture the privacy features in the application, as in Table VIII. Like in PBAC, each access behavior to private information should be attached an access purpose. The access decision is used to verify whether the access purpose complies with the intended purposes predefined on targeted objects. Such a decision process is named query compliance [8], illustrated as PC5 is derived from PC4 in Table VIII.

TABLE VIII. PRIVACY CONSTRAINTS (PC)

PC_1	If $u_k \leftharpoondown r_j$ and $r_i \leq r_j$, then $u_k \leftharpoondown r_i$;
PC_2	If $ob_k \leftharpoonup p_j$ and $p_i \preceq p_j$, then $ob_k \leftharpoonup p_i$;
PC_3	If $u_i \leftharpoondown r_j$ and $r_j \leftharpoonup p_k$, then $u_i \leftharpoonup p_k$;
PC_4	When $u_i \leftharpoondown r_k$, $<r_l, op, ob_j>$, and $u_i \rightarrow_{op} ob_j$, then $u_i \leftarrow_{op} ob_j$ if $r_l \preceq r_k$ (security base)
PC_5	When $u_i \leftharpoonup p_k$, obj $\leftharpoonup p_l$, and $u_i \rightarrow_{op} ob_j$, then $u_i \rightarrow_{op} ob_j$ if $p_l \preceq p_k$ (query compliance)

These privacy constraints can be directly implemented in database systems (e.g., as data security labels or triggers). When running the website privacy request, database security reference monitor will enforce the user access behaviors to private information, and finally, enforce the privacy features that are required by the website application.

B. *Privacy Artifacts Compliance Relations*

The compliance relationships among the above privacy artifacts are illustrated in Fig.2. Each line in the figure refers to a compliance relation from a junior artifact to a senior artifact. The compliance verification mainly includes two aspects: *consistent privacy semantics checking* and *privacy features coverage checking* in these relations. For instances, PG_1 - $HLPP_1$, indicating $HLPP_1$ complies with PG_1; PR_{11} – {$LLPP_{11}$, $LLPP_{21}$}, indicating the semantics of PR_{11} is covered by two junior subartifacts: {$LLPP_{11}$, $LLPP_{21}$}.

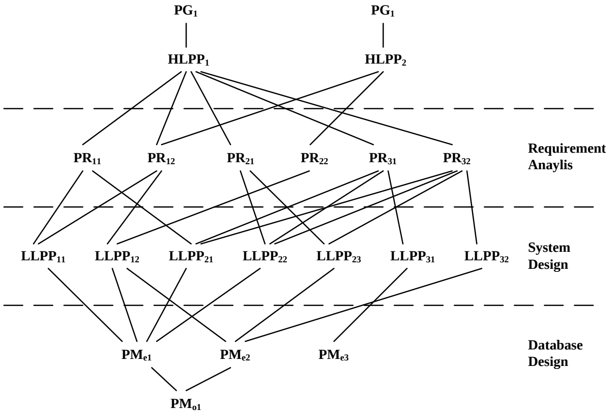


Figure 2. Privacy artifacts compliance relations

The primary private artifacts, PR, LLPP, PMe, PMo, can be introduced and refined during the system requirements analysis, software design, and database design phase of the system development, respectively. On this point, privacy aware engineering can be achieved by general software development methodologies. In fact, Fig.2 gives us a relationship trace of privacy artifacts compliance in the privacy engineering flow, which can be used as a traceability tool (e.g., traceability matrix) [13].

V. RELATED WORK

Privacy as a social and legal issue, traditionally, has been the concern of social scientists, philosophers and lawyers. However, the extended use of various software applications in the context of electronic environments sets additional technology-related principles and goals.

Technical research efforts aiming to the electronic privacy of individuals fall in two main categories: security-oriented requirement engineering methodologies and privacy enhancing technologies [14,19,21,22]. The former focus on methods and techniques for considering security/privacy issues during the early stages of system development and the latter describe technological solutions/mechanisms for assuring user privacy during system implementation. Existing privacy enhancing technologies are classified in six categories, namely: (1) privacy policy administrative tools, (2) privacy-aware information tools like privacy preserving database management systems, (3) anonymizer tools, data releasing services and architectures for data sharing, (4) divers pseudonymizer tools for user identity management, (5) track and evidence erasers, and (6) data and communication encryption tools. An overview of these categories can be found in [12, 18].

The main limitation of current privacy engineering methodologies is that they do not link the identified requirements with their implementation solutions. Another limitation is that little or no research has so far taken place in order to address requirements for software systems to which privacy artifacts compliance applies. This lack of knowledge makes it difficult to determine which software solution best fits the organizational privacy compliance needs [19]. To achieve all privacy goals during system development, it is necessary to improve the alignment and verification from privacy policies and requirements to implementation constraints, i.e. the privacy compliance engineering process. Several authors have proposed privacy design frameworks for specific domain: Feigenbaum et al. proposed privacy engineering guidelines for digital rights management systems [11], while Hong et al. proposed privacy risk models as an approach to the design of privacy sensitive [18]. In recent years, privacy compliance in the life cycle has been invested. Kalloniatis et al. have proposed the PriS method, a security requirement engineering method which incorporates privacy requirements early in the system development process, to addressing privacy requirements in system design [19]. The PriS method considers privacy requirements as organizational goals that need to be satisfied and adopts the use of privacy-aware process patterns. Spiekermann et al. [22] have introduced a three-sphere model of user privacy concerns and related it to system operations (data transfer, storage and processing) and given two types of approaches to engineering privacy: “privacy-by-policy” and “privacy-by-architecture.” The privacy-by-policy approach focuses on the implementation of the notice and choice principles of fair information practices, while the privacy-by-architecture approach minimizes the collection of identifiable personal data and emphasizes the anonymization and client-side data storage and processing.

For the privacy compliance implementation, privacy requirements can be directly transformed to a set of constraints of permissions and roles [8, 17], which is more straightforward. But it needs the logic database design before the transformation. The verification of privacy constraints to requirements is a both technique and engineering work in privacy engineering. Haubner et al. propose a formal task-based privacy model [16], which is defined as a state machine model. Privacy compliance requirements can be verified based on such a state machine model.

VI. Conclusion And Future Work

In this work we aim to present a holistic view of the privacy aware engineering in which we derive privacy models/constraints from accepted privacy principles as well as from privacy goals (user concerns), and propose a privacy engineering flow that integrates various involved privacy artifacts in order to provide engineers a clear roadmap for building privacy-friendly information systems.

We define the notion privacy aware engineering as an engineering process that systematically integrates those privacy artifacts emerging during software development in a “better" alignment. The privacy engineering flow is proposed as a general preparation for privacy engineering in most privacy-enhancing applications. With a case study, we specify the privacy artifacts involved in the flow, discuss some efficient privacy models and mechanisms, and analyze their compliance verification.

The work is in its infancy, with a number of fascinating research challenges waiting to be addressed. These challenges range from defining languages and models for various artifacts [4], to defining methodologies for *compatibility* and *compliance* privacy artifact transforming, to even methods for translating privacy artifacts into internal data structures that can be used for automated privacy management support at design time or runtime. In future, we will continue the privacy artifacts compliance verification work and try to build a tool to automatically accomplish this task.

Acknowledgment

This work was supported by NSFC 60673140 and NHTP (2007AA01Z156, 2008ZX01045, 2009CB320706).

References

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, Hippocratic databases. In Proc. of the VLDB. Hong Kong, China, 2002.

[2] S. Ali, J. Hall, Introducing regulatory compliance requirements engineering, Securing Electronic Business Processes, 439-447, Vieweg, 2006.

[3] R. Ananthanarayanan, A. Gupta, and M. Mohania, Towards automated privacy compliance in the informationlife cycle, LNCS 4891, pp. 247–259, 2008.

[4] A. I. Antón, E.Bertino , N. Li, T. Yu.: A roadmap for comprehensive online privacy policy management,. Commun. ACM 50(7), 109–116 (2007).

[5] A. I. Antón, D. Bolchini, and Q. He. The use of goals to extract privacy and security requirements from policy statements. Proc. of the 26th IEEE Int'l Conf. on Software Engineering (ICSE'03), 2003.

[6] A. I. Antón, J. B. Earp, and A. Reese. Analyzing web site privacy requirements using a privacy goal taxonomy. Proc. of the 10th IEEE Joint Requirement Engineering (RE) Conf., 2002

[7] T. D. Breaux and A. I. Antón. Analyzing regulatory rules for privacy and security requirements. IEEE Ttransactions on software engineering, 34(1), 5-20, 2008.

[8] J.-W. Byun, E. Bertino, and N. Li. Purpose-based access control of complex data for privacy protection. Proc. of SACMAT, 2005.

[9] R. Crook, D. Ince, and B. Nuseibeh. On modelling access policies: Relating roles to their organisation context. Proc. of the 13th IEEE Conf. on Requirements Engineering, 2005.

[10] G. Denker, and D. Martin, Using rule to define the semantics of privacy policies, www.cls.sri.com/user/denker.

[11] J. Feigenbaum, M. J. Freedman, T. Sander, and A. Shostack, Privacy engineering for digital rights management systems, LNCS 2320, 2002

[12] S. Fischer-Hbner. It-security and privacy – design and use of privacy-enhancing security mechanisms. LNCS 1958, 2001.

[13] O. C. Gotel and A. C. Finkelstein. An analysis of the requirements traceability problem. Proc. of the 1st Int'l Conf. on Requirements Engineering, 1994.

[14] P.Guarda and N. Zannone. Towards the development of privacy-aware systems. Information and Software Technology(2009) 2:337-350.

[15] C. B. Haley, R. Laney, J. D. Moffett, and B. Nuseibeh. The effect of trust assumptions on the elaboration of security requirements. Proc. of the 12th IEEE Int'l Requirements Engineering Conf., 2004.

[16] S. F. Haubner and A. Ott. From a formal privacy model to its implementation. Proc. of the 21st National Information Systems Security Conf., Arlington, VA, Oct. 1998.

[17] Q. He. Requirements-based access control analysis and policy specification, Ph.D Dissertation, North Carolina State University, 2005.

[18] J.I. Hong, J. Ng, S. Lederer, J.A. Landay,Privacy risk models for designing privacy-sensitive ubiquitous computing systems, Proc. of the 5th Conf. on Designing Interactive Systems, Boston MA, August 2004.

[19] C. Kalloniatis, E. Kavakli, and Stefanos Gritzalis, Addressing privacy requirements in system design: the PriS method, Requirements Eng (2008) 13:241–255.

[20] S. Kenny, J. Borking, The value of privacy engineering, Journal of Information,Law and Technology 4 (1) (2002).

[21] M. N. Kreeger and I. Duncan. Engineering secure software by modelling privacy and security requirements. Proc. of the 39th International Carnahan Conf. on Security Technology, 2005.

[22] S. Spiekermann and F. Cranor, Engineering privacy, IEEE Ttransactions on Software Engineering, Vol. 35, 2009.