Privacy Requirement Modeling and Verification in Cloud Computing

Jin, Wang
College of Computer Science and Technology
Nanjing University of Aeronautics and Astronautics
Nanjing, China
woodenwang55@hotmail.com

Abstract—**Cloud computing, the architecture which shares dynamic heterogeneous characteristics in the cross-layer service composition, has affected traditional security, trust and privacy mechanisms which are mainly based on data encryption and access control. Approaches that can support accurate privacy requirement description and verifiable compliance between the privacy requirement and system practice need to be developed to fit this new paradigm. To tackle the issues of privacy requirement modeling and verification in cloud computing, a framework that supports model checking consistency, entailment and compliance with the formal definition of privacy requirements and privacy model of cloud application is proposed. This paper provides an overview of the scientific research problem, approaches to solve the problem and ways to evaluate the solution found by the research related PhD thesis.**

Keywords—Cloud computing, privacy requirement, model checking, formal model

I. PROBLEM

Scalability, on-demand access and network-based message delivery are three core characteristics of cloud computing [1]. As a scalable and hierarchical distributed collaboration paradigm, cloud computing is envisioned as an XaaS (X As a Service) architecture, combined with the advantage of reducing cost by sharing computing and storage resources [2]. Although there is a large push towards cloud computing, security and privacy are the major challenges which inhibit the cloud computing's wide acceptance in practice. A survey conducted by the US Government Accountability Office (GAO) states that "22 of 24 major federal agencies reported that they were either concerned or very concerned about the potential information security and privacy risks associated with cloud computing" [3]. Different from the traditional architecture in which users have full control of their privacy data, the internal operations of cloud computing software systems are usually transparent to the users and once the privacy data of the users are collected, they will lose control over it.

The essence of privacy protection in cloud computing is the rights and obligations of individuals and service providers with respect to the collection, use, disclosure, and retention of PII (Personally Identifiable Information) [4,5]. Providing verifiable mechanism in design phase to ensure that the practice of software is compliant with the privacy requirement is one of the most important principles in privacy related standards and regulations such as OECD [6] and ISO29100 [7]. The issues appearing in this principle are manifold.

Firstly, we need ***an approach to precisely describe the privacy requirement.*** On the one hand, due to the ambiguous and inconsistent essence, natural language is essentially unusable to depict a user concerned privacy requirement. On the other hand, directly using a specification language, such as Linear Temporal Logic (LTL) and Description Logics (DL), is too complicated to be implemented by the requirements analysts and system designers. Therefore, we must establish one privacy requirement description method that can make a balance between the expressiveness and the applicability.

Secondly, considering the multi-participant and outsource nature of cloud computing, we must guarantee there is no conflict among different participants which means ***the privacy requirement among different participants should be consistent with each other.***

Thirdly, some laws and regulations in different application contexts, such as the Children Online Privacy Protection Act (COPPA), the Health Information Portability and Accountability Act (HIPAA) and the Gramm–Leach–Bliley Act (GLBA), are proposed to enforce the privacy requirements of each participant and we must make sure ***the privacy requirements entail certain privacy regulations and laws.***

Finally, we need to ***verify the compliance between the cloud computing system practice and requirements.*** To support the verification, a formal model of privacy requirements and the specified cloud computing privacy model are needed. Moreover, to overcome the space explosion dilemma in traditional model checking, there should be some reduction method to make our verification represent real-life systems and not just toy examples.

II. RELATED WORK

A. Privacy Requirement Description

Current privacy policy definition methods can be classified into four categories, access control model based on RBAC(Role-Based Access Control), access control models using Markup Language, Semantic-web policy frameworks using description logic and declarative language with formal semantics.

For the first category, RBAC is an access control model in which access rights are specified in terms of roles. As an access control model, RBAC lacks the notion of privacy data and purpose and is insufficient for directly modeling privacy policy. Privacy-Aware Role-Based Access Control (PARBAC) model [8] notices the partial relations in roles, purpose and data subject and combined RBAC with Chandramouli's DAFMAT framework. There is no formal semantics about the relations and interactions in the model, therefore consistency and compliance checking is not feasible with it. P-RBAC [9], is the first privacy data centric RBAC expansion. However, the authors did not present any formal model for the condition and obligation except the policy in natural language which makes that policy unusable for verifying compliance and can only check the consistency in a limited context.

For access control models using Markup Language, XACML defines a general-purpose access control system and provides a privacy profile in [10]. XACML leaves obligation interpretation and rule conflict detection to the application. Therefore, the semantics of an XACML policy cannot be fully specified by the policy itself which makes the formal semantics rely on specified application and cannot do the consistency and compliance checking with the model itself. IBM proposed a language called EPAL [11] to encode privacy policies. Similar to XACML, the language is based on XML and uses a set of attributes called vocabularies. The main issue about EPAL is that the obligation definition is natural language based and hard to expand to support formal semantics. The Platform for Privacy Preferences (P3P) is a privacy language intended for use by web site operators in informing their visitors of their data practices [12]. P3P policies were not originally intended to describe the exchange of privacy data among multi participants context. In describing data collection requirements, all requirements in P3P are categorical with fixed options which makes it impossible to expand to adapt different domains and application contexts. P3P also lacks a formal semantics, which may have led to language misuse.

For Semantic-web policy frameworks, naturally, the Web Ontology Language (OWL) and its predecessors can depict classification hierarchies with data type constraints. The Description logic behind those languages is a subset of first-order logic for expressing knowledge. Some frameworks such as KAoS [13] and Rei [14] that contains an OWL policy ontology are proposed for expressing rights, prohibitions, and obligations. Eddy [15] further extends previous work with the formal model and consistency checking method. Bearing on the limitation of Description logic, these methods cannot describe temporal constraints and can hardly work with the data implementation model of a specified system to check the compliance.

For declarative language with formal semantics, May et al. [16] introduce privacy APIs, which is a logical framework that includes a language to express permissions using commands. The policies are further formalized in the Promela model-checking language and can be checked using the model checker SPIN. However, the only temporal constraints in this method are "Opt-In" and "Opt-Out" that restricts the expressiveness to a very limited scale. Barth et al. [17] propose a Linear Temporal Logic (LTL) based framework CI (Contextual Integrity) for expressing and reasoning about norms of transmission of personal information. This method describes requirements with the logic formula directly which are proven too awkward to implement by the requirement analyzer and do not have the feasibility of reality. S4P, a declarative language to express privacy preferences and policies are proposed in [18]. The method provides a formal semantics and proof rules for their language. S4P is at a higher abstraction level than our work. The purpose of S4P is to check the consistency between user preference and privacy policy.

B. Privacy Requirement Verification

Y. Li et al. [19] establish the mapping between business process execution language (BPEL) and P3P policy and verify the privacy requirements in P3P. A state machine based model is proposed and extended to monitor the compliance of privacy agreements and verify the time properties in privacy requirements at the runtime [20,21]. Barth et. al [22] use LTL formula to depict and verify privacy properties in service composition. Our team also does research on privacy modeling and verification with interface automaton and hyper-graph. These works, however, mostly focus on the SOAP web service and do not consider the hierarchical and the heterogeneous feature of cloud computing.

C. Cloud Computing Modeling and Verification

In Cloud computing environments, different types of services co-exist and collaborate with each other to offer a final system. Numerous studies have already been carried out on modeling the most common cloud computing service, Restful and SOAP services, and their combination. In Restful Service modeling, some semi-formal and formal models, such as UML state diagram, Petri Net and finite state machine, are proposed to depict the Restful service from different aspects. In SOAP service modeling, some standards, such as BPEL and WS-CDL, have been proposed to reduce the complexity required to compose web services, hence reducing time and costs, and increase overall efficiency in businesses. To verify functional and non-functional properties, these standards are further extended and formalized by transition system [29], process algebra [30] and Petri Net [31]. Some research realize the cross-layer service composition nature of cloud computing and propose some approaches to analyze different aspects ranging from resource management model [32] to information store model [33]. None of the aforementioned approaches focus on privacy analysis which makes modeling privacy properties in cloud computing still an open challenge.

III. Proposed solution

To mitigate the aforementioned challenges in Section I, an approach considering four aspects, i.e., definition, formalization, reduction and verification, respectively, is proposed.

On the privacy requirement side, the start point of our solution is to precisely describe privacy actions and constraints. Based on refinement of current privacy protection standards and regulations, we systematically analyze the partial order relation among privacy datum, roles and purposes and define the meta-model of privacy action(Fig 1) as a basis for the privacy model checking.

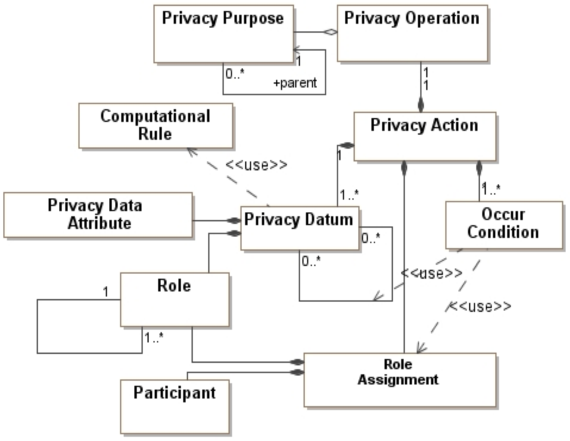


Fig 1 Privacy Action meta model.

To express the occurrence condition and obligations of privacy action, we further design three categories of event templates, EXISTENCE, BINARY RELATION and PLURAL RELATION, respectively. The declarative nature of our event templates makes them easy to use by privacy analyzers. To support the formal verification, we map each event to the corresponding LTL formula. Table 1 are some typical events and their corresponding LTL mapping. Considering the influence of partial order relations, we also give the formal definition and detection algorithm of privacy action inclusion relationship. Different from traditional LTL formulas which can monitor two or more properties simultaneously, our model is safely assumed to be a single event system. Using generated LTL formula and inclusion relationship, we can finally achieve the single event finite automaton to do the consistency and entailment checking.

Let us take one key constraint , "*Online Service does not collect personal information from any visitor prior to collecting age information",* in §312. 2 of COPPA as an example. With our language, this term can be represented by two privacy actions and one event constraint as follow (we use ***T*** to represent universal set and \ for relative complement , OS and SP are the abbreviation of Online Service and Service Provider):

p_1 = <{(User, age) },(Collect, ***T***), (Operator, User), (OS,SP),Φ>

p_2 = <{(User, PII \ age)},(Collect, ***T***), (Operator, User), (OS, SP),Φ>

e_1= Prior(p_1, p_2)

On the cloud computing system side, when services in SaaS usually use SOAP/WS-*, most products in IaaS and PaaS, such as OCCI, OpenStack, mOSAIC, Google Map, Yahoo!Local, are Restful service based. To verify the privacy requirement in cloud computing system, we need to formally define these two kinds of service and present a model can depict the multi-layer collaboration between these heterogeneous services. Firstly, a formal model that can map privacy action and atomic service request/response are stated. For SOAP service, this mapping is apparent and directly based on WSDL description. For Restful service, on the other hand, the HATEOAS (Hypermedia As The Engine Of Application State) constraint makes the internal transition more complex. To correctly represent those transitions caused by the iteration relation between resources and links, our approach defines a resource link mapping tree and then transforms that tree to the automaton. Furthermore, a cross-layer interaction model will be created based on the atomic service model and control flow.

TABLE1 EVENT TEMPLATE EXAMPLES

Template	***Description***	***LTL Mapping***
Least(*n*, *a*)	*a* will occur at least *n* times.	$\Diamond(a \wedge \bigcirc(Least(n\text{-}1,a))$
Response(*a*, *b*)	Every *a* is eventually followed by at least one *b*.	$\Box(a \Rightarrow \Diamond(b))$
MultiOptionResp (*a*, *B*)	If *a* occurs, there must be at least one privacy action in *B* occurs after *a*.	$\Box(a \rightarrow \Diamond\ (b_1 \vee b_2 \vee \ldots \vee b_n))$

One of the most severe challenges for verifying the cross-layer model is the state space explosion. The state space for 30 services composition can reach 10^6 or more which makes verifying real-life cases impossible before reduction. In our approach, both the privacy requirement and the privacy operations of the cloud computing system can be formalized as a transition system where the privacy actions are used as the transition guards. A lot of elements in privacy actions are disjoint with each other, privacy datum and participant for example. Taking advantage of these disjoint elements, we can achieve the partial model from the original formal model. For example, the privacy datum set we want to verify is { *name*, *address*, *email* }, we can obtain the sub-models for *name* , *address* and *email* respectively and just check each sub-model instead of checking the whole formal model. To generate the sub-model of one specified privacy datum, we firstly get all the privacy actions containing that privacy datum. Then we analyze the transitions that will affect or be affected by these privacy actions and finally remove those transitions which have no relation to the specified privacy datum from the original formal model.

IV. Preliminary Work

We started our work by examining the minimum privacy disclosure in SOA architecture [23]. When analyzing the state of art privacy regulations and standards, we realized most current privacy requirement definition methods lack a theoretical foundation and therefore is not amenable to verification or reasoning. Futhermore, developed from SOA, cloud computing introduces multi-layer and heterogeneous service collaboration which makes privacy protection a more complex challenge. We list most relevant up-to-date work and achieved results as below:

- We have introduced an XML based privacy requirement language which preliminarily approaches the problem of expressing temporal constraints [24]. To give the privacy requirement more precise and formal semantics, we further defined a declarative privacy policy language with its formal model [25];

- We have focused on the privacy data in SOAP and Restful service and established the formal privacy model for these two types of services [26,27] which are the basis of cloud computing;
- We have analyzed the privacy data with the predicate constraints in SOAP service composition and get a feasible path generation method that can support our future verification [28].
- We have conducted a study aimed at reducing the state space of privacy requirement and verifying the consistency among different privacy requirements [25].

V. EXPECTED CONTRIBUTIONS

We are currently working on several directions, that will address the following contributions.

From the privacy requirement modeling aspect, a privacy action meta-model and declarative requirements definition language are proposed. Constraints templates and their LTL mapping are stated to support formal specification generation. A Formal semantics of that language based on an automaton are established to support further model checking. The privacy action inclusion relationships are introduced to reduce the state space in model checking.

From the cloud computing privacy modeling aspect, the privacy model for most common service –Restful and SOAP - in cloud computing are proposed based on mapping between system behavior and privacy action. A formal model supporting multi-layer structure and heterogeneous service collaboration in cloud computing will be represented based on Restful and SOAP privacy model.

From the privacy requirement verification aspect, an automaton partition method will be presented to mediate the state space explosion issue in model checking. A series of prototype toolkits will be implemented to provide semi-automated privacy data extraction, privacy requirement definition and verification.

The main contribution can be summarized as follows.

- A declarative privacy requirement language with formal semantics
- A formal privacy model for cloud computing
- A model checking Reduction method based on transition relationship and privacy datum feature.

VI. PLAN FOR EVALUATION AND VALIDATION

We plan to validate and evaluate our work from three aspects: correctness, feasibility and performance.

At the correctness side, to measure the expressive power of our privacy requirements modeling approach, we will compare our method with other related works using the following criterias: Hierarchical data structure support, Role support, Purpose Specification support, Temporal constraints support, Privacy operation definition support, Compliance checking support, Consistency checking support and User readable. Those criterias are retrived from the OECD, ISO 29100 and other related standards. To evaluate the verification correctness, we plan to conduct an experiment using the same data in [15] and compare our conflict detection result with theirs.

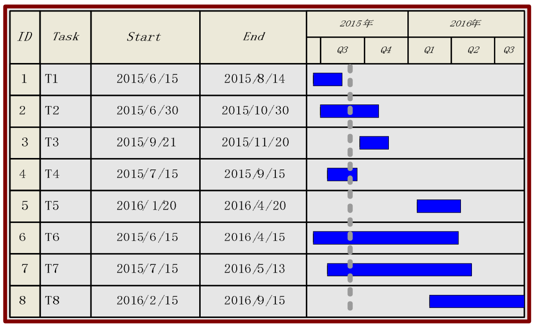
At the feasibility side, we intend to conduct a set of real case study to model COPPA, HIPAA and the policies from the related service providers to check the limitation of our privacy requirements modeling method and determine which kind of policies can be or cannot be modeled with our approach. Furthermore, we aim to model some open-source cloud computing applications which include both Restful and SOAP service interactions to analyze the compliance of the privacy requirement.

At the performance side, we intend to conduct several experiments based on the benchmark from [34] and to investigate the following indicators:

- Number of specifications
- Number of privacy datum items;
- Number of constraints
- Number of privacy action inclusion constraints;
- Number of application states in cloud system

VII. CURRENT STATUS

Currently, we are proposing a declarative privacy policy language with the formal semantics that is expected to be the input of our further consistency and entailment verification. The case study and performance evaluation will be performed in parallel with our theoretical research. To finalize our work, we identify the following tasks that will lead to a Ph.D. dissertation. The planned timeline in terms of the expected contributions and current status (dash line) is shown in Figure 2.



ID	Task	Start	End
1	T1	2015/6/15	2015/8/14
2	T2	2015/6/30	2015/10/30
3	T3	2015/9/21	2015/11/20
4	T4	2015/7/15	2015/9/15
5	T5	2016/ 1/20	2016/4/20
6	T6	2015/6/15	2016/4/15
7	T7	2015/7/15	2016/5/13
8	T8	2016/2/15	2016/9/15

Fig. 2: Planned timeline and current status

- To model the privacy requirement with the formal semantics (T1);
- To verify the consistency of privacy requirements based on reduction formal specification (T2);
- To provide a verification approach for the entailment of privacy regulation and laws (T3);
- To define mappings between cloud computing behavior and privacy action (T4);
- To study the privacy model of cloud computing that can reflect the hierarchical and heterogeneous characteristics (T5);
- To implement a semi-automated privacy framework for supporting to extract privacy, data definition from the system (T6);

- To evaluate our work by case study and performance experiment (T7);
- To finish the Ph.D. thesis and dissertation (T8).

ACKNOWLEDGMENT

This research has been supported by This work is supported in part by the National Science Foundation of China under grants (No.61272083, No.61373137, No.61373017)

REFERENCES

[1] Foster, Ian, et al. "Cloud computing and grid computing 360-degree compared." Grid Computing Environments Workshop, 2008. GCE'08. Ieee, 2008.

[2] Fox, Armando, et al. Above the clouds: A Berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28 (2009).

[3] GAO, "Information Security: Additional Guidance Needed to AddressCloud Computing Concerns," United States Government Accountability Office (GAO), October 6, 2011.

[4] Pearson, Siani. "Taking account of privacy when designing cloud computing services." Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. IEEE Computer Society, 2009.

[5] Pearson, Siani, and Andrew Charlesworth. "Accountability as a way forward for privacy protection in the cloud." Cloud computing. Springer Berlin Heidelberg, 2009. 131-144.

[6] Organisation for Economic Co-operation and Development. OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data. OECD Publishing, 2002.

[7] ISO/IEC 29100. Information technology – Security techniques– Privacy framework. ISO/IEC 29100 (1st edition), 2011.

[8] Q. He. Privacy enforcement with an extended role-based access control model, NCSU Computer Science Technical Report TR-2003-09, 2003.

[9] Q. Ni, E. Bertino & J. Lobo,"An Obligation Model Bridging Access Control Policies and Privacy Policies," in Proc. of ACM SACMAT, pp.133-142, 2008.

[10] OASIS. XACML's Privacy profile. Available: http://www.oasisopen.org/committees/document.php?document_id=37643&wg_abbrev=xacml

[11] P. Ashley et al.,"Enterprise Privacy Authorization Language (EPAL)," Research Report RZ 3485, IBM Research, 2003.

[12] Cranor L et al, Platform for privacy preferences (P3P) specification. W3C working group note,2006

[13] Uszok A, Bradshaw JM, Lott J, Breedy M, Bunch L. New developments in ontology-based policy management: increasing the practicality and comprehensiveness of KAoS, In: IEEE workshop on policies for distributed systems and networks,pp 145–152, 2008.

[14] Tonti G, Bradshaw JM, Jeffers R, Montanari R, Suri N, Uszok A. Semantic web languages for policy representation and easoning: a comparison of KAoS, Rei, and Ponder, LNCS 2870:419–437, 2003.

[15] T. D. Breaux, H. Hibshi, and A. Rao. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements, Requirements Engineering, pages 1–27, 2013.

[16] May MJ. Privacy APIs: formal models for analyzing legal and privacy requirements, Ph.D. Thesis, University of Pennsylvania,2008.

[17] A. Barth, A. Datta, J. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: Framework and applications, In IEEE Symposium on Security and Privacy, pages 184-198, 2006.

[18] M. Y. Becker, A. Malkis, L. Bussard, S4P: A generic language for specifying privacy preferences and policies, Technical Report MSR-TR-2010-32, Microsoft Research, 2010.

[19] Y. Li, S. Benbernou, H. Paik, and B. Benatallah. Formal consistency verification between bpel process and privacy policy. Proc of Privacy Security Trust PST'2006. New York, NY, USA : ACM 2006 : 212–224.

[20] Salima Benbernou, Hassina Meziane, Mohand-Said Hacid. Run-Time Monitoring for Privacy-Agreement Compliance. LNCS 4749 : Proc of ICSOC 2007, Berlin: Springer,2007: 353-364.

[21] Karima Mokhtari, Salima Benbernou, Mohand-Said Hacid, Emmanuel Coquery, Frank Leymann. Verification of Privacy Timed Properties in Web Service Protocols. Proc of IEEE SCC 2008, Washington: IEEE 2008: 593-594.

[22] Adam Barth, John Mitchell, Anupam Datta, and Sharada Sundaram. Privacy and utility in business processes. Proc of Computer Security Foundations Symposium, Washington: IEEE, 2007:279–294.

[23] Liu, Linyuan, Haibin Zhu, and Zhiqiu Huang. "Analysis of the minimal privacy disclosure for web services collaborations with role mechanisms." Expert Systems with Applications 38.4 (2011): 4540-4549.

[24] Lu Jiajun, Huang Zhiqiu, Wang Jin, et al. Behavior-oriented privacy policy description for Web services composition. Journal of Frontiers of Computer Science and Technology,2013,7(7) : 592-601.

[25] Wang Jin, Huang Zhiqiu. Privacy Requirement Modeling and Consistency Checking in Cloud Computing. Journal of Computer Research and Development. 2015, 52(10).

[26] Cai Zheng-ping, Huang Zhi-Qiu, Wang Jin, et. al. Research of Web Services Composition Transaction Coordination Framework based on BPEL and WS-TX. Computer Science. 39.6 (2012): 120-124.

[27] Tie Wei, Huang Zhi-Qiu, Wang Jin. BPEL based asynchronous interaction and composition of RESTful Web Service. Computer Engineering & Science. 35.4 (2013): 29-36.

[28] Wang Jin, Huang Zhi-qiu ,Tang Jiajun, et. al. Predicate Constraint Oriented BPEL Modeling and Feasible Path Analysis. Journal of Computer Research and Development. 2015, 51(4): 838-847.

[29] Foster H, Uchitel S, Magee J, et al . Model-based verification of web service compositions. Proc of 18th IEEE Int Conf on Automated Software Engineering. Piscataway, NJ: IEEE, 2003: 152-163

[30] Ferrara A. Web services: A process algebra approach. Proc of 2nd ACM Int Conf on Service Oriented Computing. Yew York : ACM,2003: 242-251

[31] Ouyang Chun, Verbeek Eric, Wil M.P, et al. Formal semantics and analysis of control flow in WS-BPEL . Science of Computer Programming,2007,67(2/3), 162-198.

[32] De Boer, Frank S., et al. "Formal modeling of resource management for cloud architectures: An industrial case study." Service-Oriented and Cloud Computing. Springer Berlin Heidelberg, 2012. 91-106.

[33] Fitch, Daniel F., et. al. "A Petri Net Model for Secure and Fault-Tolerant Cloud-Based Information Storage." SEKE. 2012.

Westergaard, M.: Better Algorithms for Analyzing and Enacting Declarative Workflow Languages Using LTL, In: Rinderle, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, 2011.