HAL
archives-ouvertes.fr

Usage Control Enforcement - A Survey

Åsmund Nyre

▶ **To cite this version:**

Åsmund Nyre. Usage Control Enforcement - A Survey. 1st Availability, Reliability and Security (CD-ARES), Aug 2011, Vienna, Austria. pp.38-49, 10.1007/978-3-642-23300-5_4 . hal-01590390

HAL Id: hal-01590390
https://hal.inria.fr/hal-01590390

Submitted on 19 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

Usage control enforcement - a survey

Åsmund Ahlmann Nyre

Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
nyre@idi.ntnu.no

Abstract. Sharing information allows businesses to take advantage of hidden knowledge, improve work processes and cooperation both within and across organisations. Thus there is a need for improved information protection capable of restricting how information is *used*, as opposed to only accessed. Usage Control has been proposed to achieve this by combining and extending traditional access control, Digital Rights Management and various encryption schemes. Advances in usage control enforcement has received considerable attention from the research community and we therefore believe there is a need to synthesise these efforts to minimise the potential for overlap. This paper surveys the previous efforts on providing *usage control* enforcement and analyses the general strengths and weaknesses of these approaches. In this paper we demonstrate that there are several promising mechanisms for enforcing usage control, but that reliable empirical evidence is required in order to ensure the appropriateness and usability of the enforcement mechanisms.

1 Introduction

Despite almost daily reports of security flaws and breached security, Internet users continue to share information at an increasing rate. Basically about anything from personal habits to sensitive corporate secrets, using a wide variety of communication channels. The fundamental problem is however that current security systems are unable to enforce any restrictions on the access, use or distribution of information once it has been transferred from one system to another. Hence, the sender involuntarily looses any control of the information and must resort to trusting the receiver to not misuse information. It is this problem that has lead to serious concerns about users privacy on the Internet, and the music and film industry's concern about intellectual property rights violations, and corporations concern about sensitive information misuse.

Usage control have been proposed as a means to remedy this problem by extending common security mechanisms beyond single systems such as PCs, servers or entire corporate systems. The idea is to provide a model for expressing and enforcing restrictions on how the information is to be *used* . Current mechanisms such as access control, Digital Rights Management, confidentiality and privacy protection all attempt to restrict information in one way or the other. The focus

of usage control is to create an holistic approach to restricting information, such that it may be used for any of the purposes listed above.

The purpose of this review is to identify and analyse existing mechanisms for usage control enforcement in order to identify shortcomings and possible improvements.

The remaining parts of the paper is organised as follows. First, in Section 2 the research method employed is outlined and discussed. Next, in section 3 we provide a brief introduction to usage control. We review enforcement mechanisms in Section 4 and in Section 5 we report on the testing and evaluation that has been conducted . Section 6 includes our analysis and discussion of the enforcement mechanisms. Related work is outlined in Section 7, before our final conclusions are given in Section 8.

2 Research Method

The research method employed in this review is inspired by the recommendations by Kitchenham and Charters [15] regarding systematic literature reviews in software engineering, however with softer requirements regarding rigour.

This review will address the following research questions:

RQ1. What usage control enforcement mechanisms have been proposed and how do they relate to each other?
RQ2. What evidence exists supporting their appropriateness?

The purpose of this review is therefore to synthesise the previous research efforts on usage control and distributed enforcement strategies to identify open issues and prevent duplication of work. To this end, we also investigate the supporting evidence of appropriateness.

The papers subject to the review were selected based on search through the main online portals of scientific publications. These portals were: IEEE Xplore, ACM Digital Library, Scopusand SpringerLink. Potential papers were selected based on a search for "usage control" and "enforcement" in title, keywords and abstracts, for each of the portals listed above. The search capabilities of these portals vary considerably, hence minor modifications have been conducted. Most notably for SpringerLink, the amount of results returned made it necessary to conduct a nested search of the two phrases.

The search strategy clearly results in several irrelevant papers being potential subject to the review. Additional criteria were therefore specified to ensure only relevant papers be included in the actual review. These criteria are referred to as inclusion and exclusion criteria, indicating both a positive definition and a negative definition of what the review should and should not contain. Published peer-reviewed papers were included in the review if meeting any of the following criteria:

- Papers presenting mechanisms for enforcement of usage control models.
- Papers reporting on testing and experience of use with such mechanisms.

The criteria may be met by only parts of the paper, thus it need not be the main focus of the paper to be included. Papers meeting one or more of the of the following criteria were excluded:

- Position papers identifying threats and challenges to usage control.
- Papers focusing solely on subsets of usage control, such as access control and digital rights management.

Note that inclusion criteria was given a higher order than exclusion criteria, such that any paper satisfying both were included in the review.

From each of the included papers, the following content was extracted and analysed:

- The main idea of the model (RQ1.).
- The main strengths and limitations of the approach (RQ1.).
- The supportive evidence of its appropriateness (RQ2.).

While the data extraction part is mainly about documenting previous efforts, the analysis part was intended to look beyond specific claims to make more general assessments of the quality and usability of the proposed enforcement mechanisms.

3 Usage control

The UCON (Usage Control) model was proposed by Park and Sandhu [27] to alleviate many of the shortcomings of existing access control mechanisms, particularly for distributed assets. The authors focus on three important parts of the model, which are *Authorizations*, *oBligations* and *Conditions*, resulting in what they denote the UCON_{ABC} model. The main elements of this model are

Subject an entity with a set of attributes (subject attributes) either holding or exercising rights on an object.

Object an entity with a set of attributes (object attributes) that a subject hold or exercise a right on.

Right a privilege held by a subject to perform certain functions on an object.

Authorization a functional predicate to be evaluated in order to decide whether the subject is allowed to perform the requested rights on the object.

oBligation a functional predicate to verify requirements a subject has to perform before or during a usage exercise.

Condition a functional predicate to verify the requirements for the environment or system to be present before or during a usage exercise.

The family of models have later been expanded and detailed using various formalisms. A survey by Lazouski et al. [20] provide a good overview of these efforts. One of the main concepts of UCON is the view on continuous enforcement (see Figure 1). Contrary to common access control mechanisms, UCON assumes authorisations to be an ongoing activity such that misbehaviour may result in real-time revocation of rights. Park and Sandhu defined 16 basic UCON models based on the different steps in Figure 1.

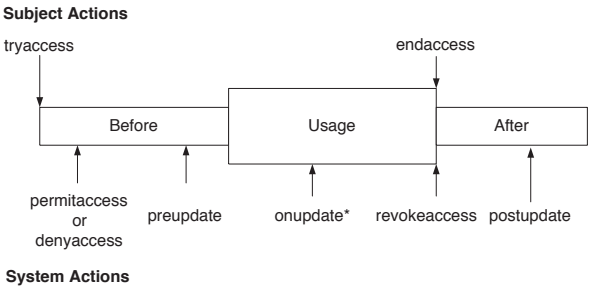

Fig. 1. UCON model with continuity of enforcement and mutability of properties[32]

4 Enforcement mechanisms

In this section we describe the identified usage controlled enforcement mechanisms. We have separated them into three main categories which are proactive enforcement, reactive enforcement and trust-based enforcement. Although these categories are not mutually exclusive, they are often treated separately.

4.1 Proactive enforcement

In this section we elaborate on the mechanisms that are mainly focused on preventing policy violation through proactive security measures. We categorise the mechanisms into three groups of client-side enforcement: trusted computing-based enforcement, hardware-based enforcement and software-based enforcement. Finally we also provide a category for server-side enforcement. Note that some of the approaches falls into several categories, but are placed in the one that best describe the focus of the mechanism.

Trusted computing Trusted computing has been introduced as one of the main building blocks to enforce usage control policies on remote client devices. Although there are many that assume the use of trusted computing or a trusted platform, we include here only the approaches where trusted computing is essential to the mechanism.

Sandhu et al. propose a Trusted Reference Monitor (TRM) residing in user space to enforce policies [31, 33]. A hardware-based Trusted Platform Module (TPM) is envisioned to provide a root of trust that together with a Public Key Infrastructure (PKI) may be used to attest cooperating platforms. The protocol between applications and TRM, as well as between different TRMs is based on challenge-response. Any request is followed by a challenge from the receiver. The requester subsequently attest the platform, application or environment through the means of a digital signature. Abie et al. propose a somewhat similar strategy based on a TPM to enforce usage policies on digital objects [1, 2]. Self-enforcing objects (SEOs) are used as a secure container to transfer objects and policies and

are also capable of enforcing the attached policy autonomously on any trusted platform.

A practical example of such an enforcement mechanism is given by Kyle and Brustolini [19]. UCLinux is a Linux Security Module that provides attestation, sealing and usage control support. Applications is provided access to the file if they are part of the trusted computing base and the file metadata contain a hash of the application. If the application is deemed trusted, UCLinux will handle any usage control enforcement to the application itself. Untrusted applications will however not be able to read nor write to the file. Alam et al. propose a similar enforcement mechanism [3] but also describe the concept of platform attestation [4]. Prior to releasing information to an authorised requester, the system verifies through a WS-Attestation procedure that the receiving platform will behave properly. A prototype implementation indicates an introduced delay of approximately five seconds, compared to not having such attestation.

Hardware enforcement Hardware UCON Engine (HUE) [25] is another enforcement mechanism based on trusted computing that unlike most mechanisms does not rely on a built-in Trusted Platform Module (TPM). Instead, HUE is designed as a secure co-processor with a designated software stack to provide integration with the operating system. While obviously more flexible and efficient than TPM-based systems, the downside is that it the tailor-made hardware is not commercially available.

Software-based enforcement The RightsEnforcer [24] is a product for enforcing simple usage control restrictions such as limiting the ability to view, print, copy and store. The mechanism is integrated in an e-mail client such that whenever a usage controlled object is sent to a receiver, the RightsEnforcer encrypts the content and sends the terms of use to a centralised RightsServer. The receiver is forced to use the RightsEnforcer to be able to decrypt the content and therefore the policy is always enforced.

Brustolini et al. propose a simplified version where policies are specified through allowing or denying specific operations (open, copy, print, etc.) on digital objects [6, 9]. The system comprises a kernel level security module for SELinux, a web browser plugin and a modified web server that offers the ability to post information objects to a web service. Upon uploading documents, users are given the option to specify any usage restrictions. When downloading content from the service, the client warns the user of unacceptable policies.

A somewhat similar strategy is employed by Kumari et al. to provide usage control enforcement for web applications [18]. The solution is based on a web browser plugin that intercepts requests for enforcement enabled web applications and subsequently enforces the obtained policy. They demonstrate the feasibility of the approach through a modified social network application that generates policies whenever a content request is made by a user. Users can tag friends with a trust level in intervals from 0 to 1 and similarly tag content with a sensitivity on the same scale. A permission p is then computed as the product $p(t, s) = t(1 - s)$ where t is the trust value of the subject and s the sensitivity

of the content. These permissions are grouped into permission classes in five intervals such that the users may define which permission classes are allowed to perform events such as "view", "copy", etc. Any request not originating from the browser plugin is treated as permission class 0 (minimum permissions). Although currently not implemented, the authors argue that to prevent users from evading enforcement, some TPM based mechanism should be in place to guarantee that the browser plugin is not hijacked by other malicious plugins.

Server-side An enforcement architecture for server-side enforcement is given in [14]. The focus of the approach is to enforce post usage obligations on possibly remote systems. Obligations may be placed on the system or on the subject (of usage) and relates to either controllable or uncontrollable objects as seen by the system. Katt et al. argue that system obligations on controllable objects, denoted trusted obligations, do not need a fulfilment check, since the system is assumed to be trusted. However for any other untrusted obligations, the system translates the obligations into a sequence of system operations including a condition at the end. Thus, if the condition holds, the obligation is assumed to be fulfilled. In the event that the obligation is not fulfilled (e.g. a file is not deleted), the system will issue any necessary compensating action as specified in the policy.

Layered UCON (L-UCON) provides partial (or layered) usage of resources. Since digital resources often are provided in a layered fashion (e.g. the varying pixel resolution of images), L-UCON provides the user with the ability to negotiate both content and policies with the content owner. Hence, the content owner may adjust a policy based on for example the usage purpose stated by the user.

Gheorghe et al. [11] utilise the Message Service Bus of Service-Oriented Architectures (SOA) to enforce business and regulatory usage control policies on messages exchanged between services. Since all messages are placed on the service bus, messages may easily be intercepted and validated for compliance with stated policies. However, policies are assumed to be fairly static and message level usage policies are not handled by the architecture.

4.2 Reactive enforcement

Reactive security measures may be seen as an acceptance of failure, since security breaches cannot be prevented, penalties are our only options. While they do seem weaker at first glance, the main reason for utilising reactive measures is lack of control. It is difficult to prevent policy violations from occurring when the enforcer has no or limited control over the target device. Since detecting such violations after the fact are in many cases easier, reactive enforcement may provide a powerful complement to any proactive enforcement. In the following we survey some of the main efforts on reactive enforcement of usage control policies.

Audit-based enforcement The approach taken by Corin et al. is not to prevent policy violation but to ensure that users are held accountable for their actions through a proper auditing process [8]. Time stamping and signature mechanisms

are used to establish evidence of communication and to bind a usage policy to the communicated objects. Then, whenever an auditing process is started by any of the communicating parties, the other parties are obliged to provide evidence that their usage is in accordance with the policies agreed upon. The communication protocol with timestamps assures that honest parties do have such evidence.

Pretschner et al. [28] argue that many high-level obligations may not be controllable by the content provider. They therefore separate between controllable and uncontrollable obligations, and further introduce the weaker notion of *observable* obligations. Unobservable obligations are weakened and adapted until they become observable, and hence subject to monitoring and auditing. Therefore, rather than controlling obligations fulfilment, the content owner can observe (indications of) policy violations. Examples include watermarking content to detect unauthorised redistribution. Pretschener et al. [28] lets the server (content provider) employ traditional access control first to determine applicable rules and authorisations. In the next step, the consumer is provided with the set of provisional actions that must be taken prior to access and the obligations that must be fulfilled during or after usage of the content. Subsequently the consumer must provide evidence that the provisional actions have been taken and that he has committed himself to the obligations and compensations. Compensations, or sanctions as they are termed, is also proposed formally in [5]. The authors show how a system may violate certain rules in a policy and still be compliant with the policy through fulfilling a set of sanctions. The sanctions are specified for a rule in the policy and therefore also known to the user prior to a policy violation. APPLE (A Posteriori PoLicy Enforcement) is based on the same principles but utilises Auditing Authorities to automatically check the policy adherence of users [10]. The Authorities either selects documents or objects they have become aware of, or selects one randomly. While some policy violations may go undetected, the authors argue that the possibility of being caught and held accountable will in some cases deter significant violations [10].

Cederquist et al. [7] attempt to handle the problem of administrative policies in usage control. That is, who is authorised to specify and change the policy guarding a specific piece of information. In a fully decentralised system based on discretionary policies this is next to impossible, especially since it may be difficult to assess who is the owner of data after multiple changes by others. The solution by Cederquist et al. is to require that users are accountable for their actions. Such that if Alice changes a policy regarding a document before sending it to Bob, then Alice is held accountable for the change. In the event that the change is unauthorised, Bob will be able to prove that he is not to blame for the unauthorised change. Thus, the basic action required on the receiver side is to ensure that the sender is accountable. The authors also point out that as a consequence, users may perform deliberate policy infringements while accepting full responsibility for it.

Monitoring Hilty et al. [12, 30] propose an active monitoring approach where the client-side monitoring component signals to a server-side enforcement mechanism whenever obligations have been fulfilled. Hence, if information some infor-

mation should be deleted within 30 days, the monitoring component signals to the provider at time of deletion and ends the monitoring of the information object. In the event that the information is not deleted, the monitoring component signals a policy violation to the provider.

4.3 Trust-based enforcement

Trust has been extensively studied in computer science and have numerous application areas. Here we describe how trust assessments may be used for enforcement purposes.

Krautsevich et al. proposes to use trust management as means to grant usage privileges to users [17]. The idea is to utilise trust relationships and trust propagation between users and content providers to determine a usage control decision between parties that are mutually untrusted. Hence a user A with a credential issued by some other user or provider B may be given usage rights to a resource controlled by C if C trusts B.

Nyre and Jaatun also propose a trust-based enforcement strategy, but in their case the trust denotes whether the content receiver may be trusted to adhere to the usage policy [26]. Further, they specify a method to compute the probability that the receiver will enforce the policy and use this probability for content dissemination decisions.

5 Testing and evaluation

To answer the second research question, we aimed to determine what evidence are provided to support the appropriateness of the proposed enforcement mechanisms. Here we describe the quantitative testing and evaluation of the enforcement mechanisms we have described. We do not consider qualitative self-assessments as evidence of appropriateness.

Performance analysis Performance analysis is the predominant form of testing strategy employed and primarily aimed at identifying latency introduced by the enforcement mechanisms.

In [11] the introduced round-trip latency said to be around 5000 μm of which the policy decision takes about 22 μm. Although the policy decision time is insignificant compared to the round-trip time, the lack of a neutral benchmark makes it difficult to assess wether the entire round-trip time is acceptable or not. Similar shortcomings are found in [13] where the read and write speeds are analysed for files of varying sizes and in [4] where the attestation request is said to introduce a a latency of about five seconds. Since these values are not benchmarked to a system without the implemented enforcement mechanism, it is difficult to assess the appropriateness.

Djalaliev et al. do provide a benchmark of regular HTTP and TLS traffic and show that the modified TLS used for attestation only introduces an increase in CPU usage of about 20% [9]. The latency analysis demonstrates that usage

controlled file retrieval introduces a penalty varying from 100% to 30% with increasing file sizes.

The UCLinux implementation in [19] is analysed to find the system boot latency and general usage latency. General usage latency was simulated through measuring the time needed to compile a Linux kernel since this involves several different file operations. The tests show that the boot process takes 9% longer to complete on UCLinux compared to regular Linux, whereas the file operations require some 10% more time to complete.

Usability testing Brustoloni et al. [6] conduct an end-user test of their proposed system. The test seems to support their claim that performance is not significantly affected and that user awareness of unacceptable policies are considerably improved. However, there are only ten participants in the test all of which are students and only 50% have a computer science or engineering background despite the fact that the test has been designed with such professionals in mind. Further, the test seems to have no reference group and hence comparison of awareness is only done with the test group, something which could potentially bias the result. Although there is nothing to suggest that the enforcement mechanism is inappropriate, we find the low participation and the limited procedure not convincing enough for the mechanism to be deemed appropriate.

6 Analysis and discussion

In this section we attempt to analyse and generalise the strengths and limitations of the enforcement mechanisms we have described so far since space limitations prevent us from dwelling into the details of each solution.

Virtually all mechanisms for usage control assume the existence of a trusted module to ensure that enforcement cannot be evaded, yet none have discussed to any extent the practical problems of establishing and maintaining a trust infrastructure based on Trusted Computing Modules. Particularly since other considerably less complex security infrastructures, such as Public Key Infrastructures, have experienced only limited success [22]. Although the benefit of usage control may be easier to identify, most of the reasons for PKI failure described in [22] also apply to Trusted Computing. Perhaps a property-based approach to Trusted Platform Module could alleviate some of the complexity of using binary hashes of applications [16]. We do however agree that TPMs and Trusted Computing-based approaches are the only means of guaranteeing enforcement on remote devices.

There are cases in which enforcement guarantees are not necessarily required, particularly in closed systems such as business environments where the predominant threat is end-users' lack of awareness. In such circumstances, perhaps a more lightweight approach could be adopted that is technically infeasible for the average user to circumvent, but without the cost of a hardware-based trust infrastructure. It seems that the RightsEnforcer [24] is the only approach investigating along these lines.

Allowing policy violation may seem like a contradiction for a policy enforcer. Massacci [23] argues that enforcement should be *reasonable*, and therefore should allowed to be circumvented in cases where there is a just cause. If users are accountable for their action, they may only be required to justify why the violation (e.g. authorisation given orally) such that penalties may be given in retrospect. Most of the audit-based approaches do offer the violation capability, but at the same time rely on previously described compensations to be carried out. Which in essence means that only exceptions that are identified in the policy are handled by the enforcer, and therefore the rigour of the policy is not reduced.

The perhaps most striking issue we have come across is the lack of empirical evidence of the enforcement mechanisms' appropriateness. This may be due to the fact that usage control enforcement is a relatively new research field and that formal models and prototypes showing technical feasibility is a prerequisite to be able to perform proper end-user tests. However, from the effort listed here we conclude that the formal and technical basis should be well in place to allow for more user centric enforcement strategies that properly balance the usability and the security provided.

7 Related work

There have been some reviews conducted previously. From the digital rights management perspective, Liu et al. [21] have conducted a survey on DRM technology but does not treat the more general case of usage control. The review by Pretschner et al. [29] is also on DRM technology, but is more focused on alternative use of such technology including privacy and business information protection. Four different DRM technologies are analysed and compared to a set of general usage control enforcement capabilities. The idea is to see whether DRM can be used for all regular consumer side enforcement. While the evaluated systems do vary to some degree, their main limitation is the lack of enforcement after rendering. In this paper, we consider enforcement of much more expressive policies than the simple content protection mechanisms of most commercially available products and services.

A recent survey by Lazouski et al. [20] does however target usage control in general. The centre of gravity is the UCON model proposed by Park and Sandhu [27] such that other initiatives are described based on how they relate to this model. Formal models, architectures, enforcement strategies and implementations are discussed and some of the open challenges outlined. Contrary to this paper, enforcement is not considered throughly.

8 Conclusion

This paper has identified existing approaches to usage control enforcement as identified in research question RQ1. Despite the relatively short period of time since the term and concept of Usage Control was coined, there has been considerable efforts in establishing proper enforcement mechanisms. While formal

models and prototype implementations have been, and still are necessary to demonstrate the technical feasibility of usage control enforcement, more effort is required in obtaining reliable empirical data on the usefulness and usability of these approaches. Therefore we are currently unable properly address research question RQ2 to judge particular approaches as more appropriate than others.

References

1. Abie, H., Spilling, P., Foyn, B.: A distributed digital rights management model for secure information-distribution systems. International Journal of Information Security 3(2), 113–128 (11 2004)
2. Abie, H., Spilling, P., Foyn, B.: Rights-carrying and self-enforcing information objects for information distribution systems. Information and Communications Security pp. 546–561 (2004)
3. Alam, M., Seifert, J., Li, Q., Zhang, X.: Usage control platformization via trustworthy SELinux. In: Proceedings of the 2008 ACM symposium on Information, computer and communications security. pp. 245–248 (2008)
4. Alam, M., Zhang, X., Nauman, M., Ali, T.: Behavioral attestation for web services (BA4WS). In: Proceedings of the 2008 ACM workshop on Secure web services. pp. 21–28 (2008)
5. Brunel, J., Cuppens, F., Cuppens, N., Sans, T., Bodeveix, J.: Security policy compliance with violation management. In: Proceedings of the 2007 ACM workshop on Formal methods in security engineering. pp. 31–40 (2007)
6. Brustoloni, J.C., Villamarín-Salomón, R., Djalaliev, P., Kyle, D.: Evaluating the usability of usage controls in electronic collaboration. In: Proceedings of the 4th symposium on Usable privacy and security. pp. 85–92 (2008)
7. Cederquist, J., Corin, R., Dekker, M., Etalle, S., den Hartog, J., Lenzini, G.: Audit-based compliance control. International Journal of Information Security 6(2), 133–151 (03 2007)
8. Corin, R., Galindo, D., Hoepman, J.H.: Securing data accountability in decentralized systems. On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops pp. 626–635 (2006)
9. Djalaliev, P., Brustoloni, J.C.: Secure web-based retrieval of documents with usage controls. In: Proceedings of the 2009 ACM symposium on Applied Computing. pp. 2062–2069 (2009)
10. Etalle, S., Winsborough, W.H.: A posteriori compliance control. In: Proceedings of the 12th ACM symposium on Access control models and technologies. pp. 11–20 (2007)
11. Gheorghe, G., Mori, P., Crispo, B., Martinelli, F.: Enforcing ucon policies on the enterprise service bus. In: On the Move to Meaningful Internet Systems, OTM 2010, pp. 876–893. LNCS, Springer Verlag Berlin (2010)
12. Hilty, M., Pretschner, A., Basin, D., Schaefer, C., Walter, T.: Monitors for usage control. Trust Management pp. 411–414 (2007)
13. Hu, H., Li, H., Feng, D.: L-ucon: Towards layered access control with ucon. In: Proceedings of the International Conference on Computational Science and Engineering. vol. 2, pp. 823 –829 (aug 2009)
14. Katt, B., Zhang, X., Breu, R., Hafner, M., Seifert, J.: A general obligation model and continuity: enhanced policy enforcement engine for usage control. In: the ACM symposium on Access control models and technologies. pp. 123–132 (2008)

15. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. EBSE Technical Report EBSE-2007-01, Keele University and University of Durham (2007)
16. Korthaus, R., Sadeghi, A., Stüble, C., Zhan, J.: A practical property-based bootstrap architecture. In: Proceedings of the 2009 ACM workshop on Scalable trusted computing. p. 29–38 (2009), ACM ID: 1655114
17. Krautsevich, L., Lazouski, A., Martinelli, F., Mori, P., Yautsiukhin, A.: Usage control, risk and trust. In: Trust, Privacy and Security in Digital Business, vol. 6264, pp. 1–12. Springer (2010)
18. Kumari, P., Pretschner, A., Peschla, J., Kuhn, J.: Distributed data usage control for web applications: a social network implementation. In: Proceedings of the first ACM conference on Data and application security and privacy. pp. 85–96 (2011)
19. Kyle, D., Brustoloni, J.: Uclinux: a linux security module for trusted-computing-based usage controls enforcement. pp. 63–70 (2007)
20. Lazouski, A., Martinelli, F., Mori, P.: Usage control in computer security: A survey. Computer Science Review 4(2), 81–99 (2010)
21. Liu, Q., Safavi-Naini, R., Sheppard, N.P.: Digital rights management for content distribution. In: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21. pp. 49–58 (2003)
22. Lopez, J., Oppliger, R., Pernul, G.: Why have public key infrastructures failed so far? Internet Research 15(5), 544–556 (2005)
23. Massacci, F.: Infringo ergo sum: when will software engineering support infringements? In: Proceedings of the FSE/SDP workshop on Future of software engineering research. p. 233–238 (2010)
24. Matson, M., Ulieru, M.: The 'how' and 'why' of persistent information security. In: Proceedings of the International Conference on Privacy, Security and Trust. pp. 1–4 (2006)
25. Nauman, M., Ali, T.: Hue: A hardware ucon engine for fine-grained continuous usage control. In: the IEEE International Multitopic Conference. pp. 59–64 (2008)
26. Nyre, A.A., Jaatun, M.G.: A probabilistic approach to information control. Journal of Internet Technology 11(3), 407–416 (2010)
27. Park, J., Sandhu, R.: The UCON_{ABC} usage control model. ACM Transactions on Information Systems Security 7(1), 128–174 (2004)
28. Pretschner, A., Hilty, M., Basin, D.: Distributed usage control. Communications of the ACM 49(9), 39–44 (2006)
29. Pretschner, A., Hilty, M., Schutz, F., Schaefer, C., Walter, T.: Usage control enforcement: Present and future. IEEE Security & Privacy 6(4), 44–53 (2008)
30. Pretschner, A., Massacci, F., Hilty, M.: Usage control in service-oriented architectures. Trust, Privacy and Security in Digital Business pp. 83–93 (2007)
31. Sandhu, R., Zhang, X., Ranganathan, K., Covington, M.J.: Client-side access control enforcement using trusted computing and pei models. Journal of High Speed Networks 15(3), 229–245 (2006)
32. Zhang, X., Park, J., Parisi-Presicce, F., Sandhu, R.: A logical specification for usage control. In: Proceedings of the 9th ACM symposium on Access control models and technologies. pp. 1–10 (2004)
33. Zhang, X., Seifert, J.P., Sandhu, R.: Security enforcement model for distributed usage control. In: Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08. IEEE International Conference on. pp. 10–18 (2008)