Trusted Privacy Domains – Challenges for Trusted Computing in Privacy-Protecting Information Sharing

Hans Löhr[1], Ahmad-Reza Sadeghi[1], Claire Vishik[2], and Marcel Winandy[1]

[1] Horst Görtz Institute for IT-Security
Ruhr-University Bochum, Germany
{hans.loehr, ahmad.sadeghi, marcel.winandy}@trust.rub.de
[2] Intel Corporation
claire.vishik@intel.com

Abstract. With the growing use of the Internet, users need to reveal an increasing amount of private information when accessing online services, and, with growing integration, this information is shared among services. Although progress was achieved in acknowledging the need to design privacy-friendly systems and protocols, there are still no satisfactory technical privacy-protecting solutions that reliably enforce user-defined flexible privacy policies. Today, the users can assess and analyze privacy policies of data controllers, but they cannot control access to and usage of their private data beyond their own computing environment.

In this paper, we propose a conceptual framework for user-controlled formal privacy policies and examine elements of its design and implementation. In our vision, a Trusted Personal Information Wallet manages private data according to a user-defined privacy policies. We build on Trusted Virtual Domains (TVDs), leveraging trusted computing and virtualization to construct privacy domains for enforcing the user's policy. We present protocols for establishing these domains, and describe the implementation of the building blocks of our framework. Additionally, a simple privacy policy for trusted privacy domains functioning between different organizations and entities across networks is described as an example. Finally, we identify future research challenges in this area.

1 Introduction

Global connectivity and easy access to distributed applications and digital services over the Internet changed the paradigm of both business and consumer use of information. The Internet offers new opportunities to individuals, e.g., e-commerce and social network services. In addition to personal computers, mobile devices, such as smart phones, allow users to access numerous services through mobile networks from any location.

Together with the new opportunities, new security threats also developed, rapidly growing in number and sophistication. Some security threats, such as identity theft, one of the fastest growing crimes on the Internet, also can cause privacy violations [1,2]. But privacy issues are much broader: individuals frequently generate and reveal a significant amount of personal and sensitive information when they use a service such as online shopping or social networking. Even if a transaction is not personalized, it always leaves a trail that can be aggregated with other information and analyzed, potentially leading to privacy leaks. Also, as devices access networks and services, information about these accesses can be recorded.[3] The users have to trust the application provider to treat their personal data in an appropriate manner, e.g., according to best practices and regulatory requirements reflected in privacy policies. The users can read statements about privacy policies on websites, but the policies do not allow for flexibility in disclosing data necessary to access the service. There are few[4] technical means to support this kind of enforcement. Ideally, the users should be able to grant access to their sensitive information only when the systems are trustworthy and should be allowed to revoke this permission.

Technical measures in the areas of modern IT security and cryptography provide only partial solutions. Because of the inherent vulnerabilities resulting from high complexity of systems, common computing platforms require careful and attentive system administration skills, and complete protections against execution of malicious code and tampering is impossible.

In this paper, we propose a conceptual framework for user-controlled privacy policies and examine first elements of its design and implementation. The goal is to improve the current status of data and privacy protection by supporting legal measures with novel technical solutions based on Trusted Computing (TC) as described below:

- We outline a general approach to creating privacy domains, in which a guardian agent (Trusted Personal Information Wallet) manages private data according to a user-defined privacy policy (Section 2). The agent can migrate to other platforms, but only in approved trusted domains.
- We describe a simple policy that requires trusted privacy domains between different organizations and entities. We build on the idea of Trusted Virtual Domains (TVDs), leveraging trusted computing and virtualization to automatically construct privacy domains for enforcing the user's policy. We describe protocols for establishing these domains and the implementation of the building blocks of the framework (Section 3).
- Finally, we address future research challenges, analyzing currently available policy languages that cannot yet support full solutions for the reliable enforcement of user controlled privacy (Section 4).

[3] Revealing private information is sometimes necessary or unavoidable outside of the Internet (e.g., in supermarkets, due to surveillance, etc.). Although we do not study these methods to gain information about individuals, we note that the revealed information inside and outside the Internet can potentially be linked.

[4] Auditing and certification are examples for at least some technology-related methods, e.g., product evaluation according to Common Criteria or certification according to ISO 27001/27002 for information security management systems in enterprises.

2 Framework for Privacy Domains

We propose to support the enforcement of privacy policies by establishing trusted domains. These policies enable the user (individual or organization) to specify fine-grained instructions for the use of private information. As the level of online activities increases and entities or organizations with complex rules interoperate, the policies may become very complex and benefit from automatic enforcement.

The proposed architecture provides mechanisms to protect sensitive and private information across IT domains and systems. The deployment of Trusted Computing technologies for privacy protection can help achieve this goal. To ensure that private information is not re-distributed to unauthorized parties, it needs to be technically bound to only those receivers that are known to comply with the policies. Communication endpoints need to attest reliably to their compliance to specified policies.

To enforce policies, we propose a "guardian agent" for the user: a *Trusted Personal Information Wallet* that is transferable between platforms and performs "verification" of the trustworthiness of a remote IT system, i.e., compliance to a specified policy. The verification helps guarantee the enforcement of the user's privacy policy when sensitive information is transmitted. Figure 1 shows an abstract illustration of the proposed concept.

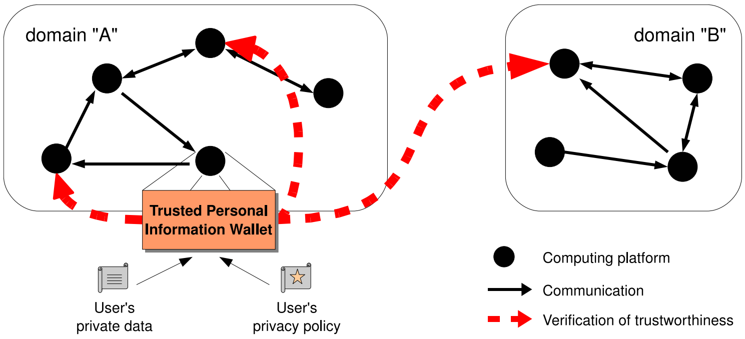

Fig. 1. Basic idea of the overall architecture

In order to achieve technical enforcement of the security and privacy policies, we develop a security architecture that allows the user to share sensitive information between computing platforms while ensuring the participating platforms have technical means to comply with the policies.

Figure 2 shows a high-level view of the process of policy enforcement. A privacy policy in a machine-readable format is incorporated into the wallet. (step 1). The wallet interprets the policy and configures security and privacy services of the underlying computing platform (step 2). The security services enforce the policy by controlling communication between applications in different domains (step 3). To reliably enforce the policy, trusted security & privacy services have

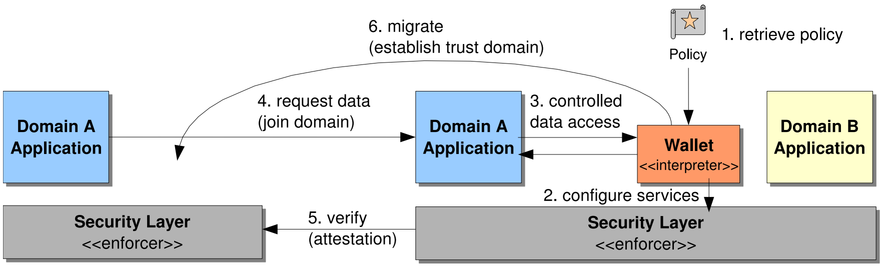

Fig. 2. Envisioned architecture for policy enforcement

to run on all participating platforms, e.g., based on a security-enhanced hypervisor [3], which allows the system owners to use legacy applications and operating systems in virtual machines, eliminating the need for new client and server side applications.

For data transmission, we propose new protocols based on existing attestation schemes of TC technology. When a user or application agent of another platform requests to access sensitive information (step 4), the security services of the source platform first verify the trustworthiness of the target platform using attestation mechanisms (step 5) to ensure the destination provides the required security mechanisms to enforce the policy. After successful verification, the wallet migrates to the destination platform (step 6) in order to act as policy decision module and to configure the security services of the target to enforce the defined policy. Service providers do not need to implement additional functionality on their server side (except for the underlying security layer) to interpret the policy or a clearinghouse for the policy interpretation. The wallet will interpret the policy and use the underlying security services of each platform to enforce it.

3 Experience with Trusted Virtual Domains

As a first step towards realizing privacy domains and policy enforcement as described before, we employ the concept of Trusted Virtual Domains (TVDs) [4,5]. In this section, we briefly review this concept and describe its novel application as privacy policy enforcement as well as our implementation of TVDs.

3.1 Concept of TVDs

A Trusted Virtual Domain (TVD) is a coalition of virtual and/or physical machines that can trust each other based on a security policy that is uniformly enforced independently of the boundaries of physical computing resources. It leverages the combination of TC and virtualization techniques in order to provide confinement boundaries for an isolated execution environment — a domain — hosted by several physical platforms.

A TVD-enforcing system supports the creation of virtual networks on physical or virtual systems. Members of a TVD can “see” and access other TVD members, but it is closed to non-members. Different instances of several TVDs can execute on the same physical platform because the underlying virtual machine monitor isolates virtual machines of different TVDs in separate compartments and isolated virtual networks.

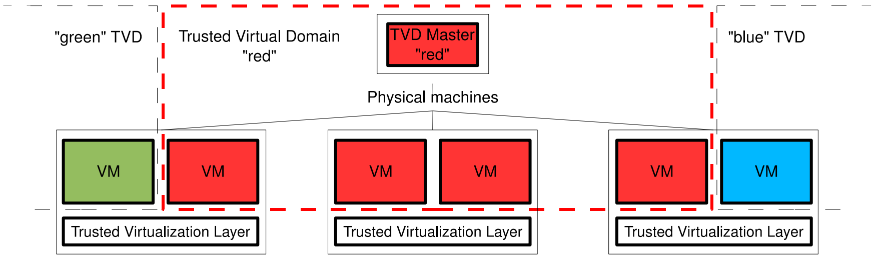
"green" TVD
Trusted Virtual Domain "red"
TVD Master "red"
"blue" TVD
Physical machines
VM
VM
VM
VM
VM
VM
Trusted Virtualization Layer
Trusted Virtualization Layer
Trusted Virtualization Layer

Fig. 3. Conceptual view of trusted virtual domains (TVDs)

Figure 3 shows an example of three TVDs (identified by colors) distributed over different physical machines. The decision whether a virtual or real machine is allowed to join the TVD is enforced based on a *TVD policy.* A special node in the TVD (*TVD Master*), e.g., implemented as a central server, controls the access to the TVD by following the admission control rules specified in the TVD policy. These rules include integrity measurements of the platforms and virtual machines that are allowed to join the domain. TC technology is used to establish trust in the reported measurements, e.g., following the Trusted Computing Group (TCG) approach, hash values of the software boot stack (BIOS, bootloader, virtualization layer as well as loaded virtual machines) are stored in and signed by a Trusted Platform Module (TPM) [6] and reported to the TVD Master during attestation. The TVD Master can reliably verify whether the reported values comply with the TVD policy and whether it can rely on the enforcement mechanisms of the local platforms.[5]

TVDs were first proposed by Griffin et al. [4] and Bussani et al. [5]. Recent research describes secure network virtualization [7], and discusses the management of TVDs in data centers [8]. The OpenTC project[6] has addressed some areas of implementing TVDs in the context of enterprise rights management and managing virtual data centers. A major issue is how the domain can be managed securely: individual machines must be able to join a domain only if they fulfill the requirements for joining, and the procedures for a platform to leave a domain must be securely constructed. These aspects of TVDs have not been studied in details yet. We describe the TVD establishment and join protocols and how TC functionality is used (see Section 3.3). The idea of applying

[5] The definition of the required integrity measurement values in the TVD policy presupposes the knowledge about the security properties of the corresponding software. In practice, trust can be achieved via independent trusted third parties that evaluate and certify IT products according to standards like Common Criteria.

[6] See `http://www.opentc.net`

the TVD concept to secure information sharing has been addressed by Katsuno et al. [9]. We extend this idea to privacy policy enforcement.

3.2 Realizing a Simple Privacy Policy with TVD

Let us consider a very simple privacy policy: only members of a particular TVD have access to the private information. The TVD policy expresses the requirements for virtual machines to join the TVD and to access this information. The TVD policy is used to implement the privacy policy, and the TVD infrastructure provides the policy enforcement for the wallet.

The wallet can act as TVD Master. In this case, it is directly responsible for policy enforcement. All parties that want to access the information have to join the TVD first. As they request to join, the wallet verifies the security properties of the joining parties using attestation. If the verification succeeds, the joining party becomes a member of the TVD and can then access sensitive information. The wallet can specify a set of "good" values for the platform configuration that are necessary to access the data.

Application scenarios for the case where the wallet is the TVD Master include those where the private information of one user is distributed to "homogeneous" data consumers, e.g., in an e-health scenario, the medical data and health records of patients are only accessible to computing platforms of medical personnel, but not to systems used by other departments.

In other classes of scenarios, where users belonging to a group want to exchange private data, it is unrealistic to have a virtual domain managed by a user's wallet. In these cases, a trusted party could provide a TVD Master responsible for policy enforcement for the group. The wallet of a user who wishes to exchange information within a group could attest the responsible TVD Master (e.g., using TCG attestation) before joining. If this attestation includes both the platform configuration of the TVD Master and the TVD policy, the wallet can ensure that information is only distributed within a TVD, where the master enforces a TVD policy that complies to the user's own privacy policy. The wallet can migrate to any node in the TVD (using conventional VM migration), and the required verification of the security properties of the destination is handled by the TVD establishment.

3.3 Implementation

Our prototype is based on the idea that a local proxy of the corresponding TVD Master, the *TVD Proxy*, is running on each physical platform that is supposed to execute virtual machines as part of a TVD. The TVD Proxy is responsible for the local enforcement of the TVD policy and performs the admission control for joining virtual machines. Since instances of multiple TVDs should be able to run isolated on one computing platform, there can be several TVD Proxies (one for each corresponding TVD) on one platform.

The main components of the trusted virtualization layer are as follows (see also Figure 4):

- *TVD-Proxy-Factory*: service that creates and manages TVD Proxies. During the establishment of the TVD, the TVD Master deploys the policy P and corresponding credentials S (cryptographic keys and certificates for, e.g., network encryption) to the TVD-Proxy-Factory. To "verify" the trustworthiness of the platform and its virtualization layer, the TVD Master requests a remote attestation of the integrity measurements, using trusted computing functionality of a TPM [6].
- *CompartmentManager*: service responsible for starting and terminating virtual machines (compartments) and taking integrity measurements of the virtual machines on start-up. This service also defines access rights for communication between active compartments.
- *TrustManager*: service providing an interface to the underlying TPM and used to create new binding keys, generate certificates for these keys, and unbind data encrypted with a binding key. The binding key is protected by the TPM and bound to the integrity measurements of the underlying platform and its trusted virtualization layer. The certificate includes these integrity measurements and permits a remote party to establish a *trusted channel* to the platform, i.e., a secure channel (providing confidentiality and integrity) bound to the integrity of the endpoint(s).

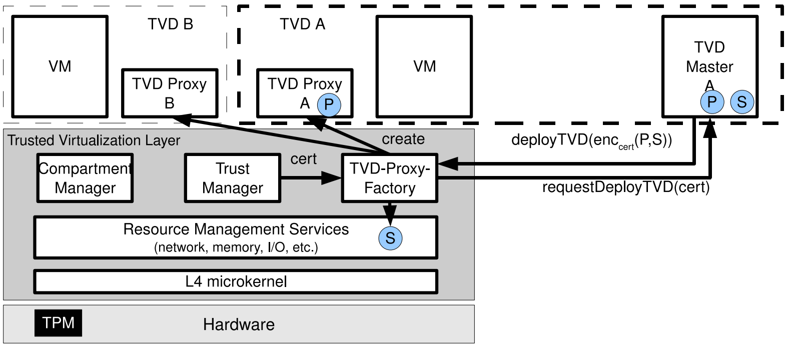

Fig. 4. TVD implementation architecture

We have implemented this design based on an existing security kernel, Turaya[7], which comprises two layers: a *hypervisor layer* based on an L4 microkernel and resource management services (memory management, I/O drivers), and a *trusted software layer* providing security services, e.g., secure storage, virtualized network, compartment management, and trusted channel establishment.

The L4 microkernel ensures isolation of processes and controls inter-process communication (IPC). Compartments can be native L4 tasks or para-virtualized

[7] `http://www.emscb.com/content/pages/turaya.htm`

Linux instances (L4Linux). Communication between compartments can be allowed or denied by applying access rights to their IPC interfaces. The microkernel enforces the IPC access control.

To support wallet functionality, it is necessary to establish a TVD and attach a virtual machine to the TVD. A TVD is established in two phases:

1. *Deploy TVD*: First, the local TVD infrastructure must be set up, including the deployment of the TVD policy and TVD credentials from the TVD Master to the trusted virtualization layer of the local platform.
2. *Join TVD*: When policy and credentials are deployed, the local TVD Proxy enforces the policy and determines if local VMs are allowed to join the TVD.

Staged establishment of the TVD was selected to avoid a central admission control that would result in considerable performance trade-offs. In this approach, the TVD policy enforcement is partially delegated to the local platforms, but the TVD Master must verify the trustworthiness (integrity state) of the platforms to establish if they can be trusted. This is done during the deployment phase.

Deploy TVD When TVD-Proxy-Factory receives a request to deploy a TVD, TrustManager generates a binding certificate $cert := (PK_{Bind}, C_{TCB})$. The TrustManager uses the TPM to generate a new binding key pair (SK_{Bind}, PK_{Bind}), where the secret key part is protected by the TPM and bound to the integrity measurement of the trusted virtualization layer (C_{TCB}). The TVD-Proxy-Factory requests deployment from the TVD Master of the desired TVD and sends the binding certificate, including the binding key PK_{Bind}.

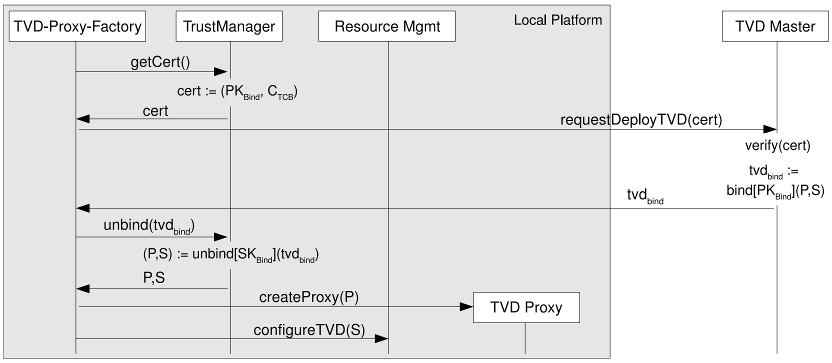

Fig. 5. TVD deployment protocol.

The TVD Master checks whether the integrity measurement of the platform matches the TVD policy. If it does, the TVD Master encrypts the TVD policy P and the corresponding TVD credentials S with the binding key PK_{Bind}, and sends the encrypted data to the local TVD-Proxy-Factory. See Figure 5.

The TVD-Proxy-Factory requests the TrustManager to unbind the data and retrieves the TVD policy and credentials (P, S). It creates a new TVD Proxy, passes the TVD policy P to it and configures the underlying resource management services (e.g., virtual network switch) with the credentials S. Now the TVD infrastructure is set up locally and ready to join virtual machines.

Join TVD The user creates the VM using the CompartmentManager. The CompartmentManager measures the integrity of the VM image (i.e., hashing the image file), stores the measurement for future requests (during runtime), starts the VM in a compartment, and returns a compartment identifier (unique during runtime of the platform). The user can request to join the compartment to the TVD by passing the compartment ID to the TVD Proxy.

The TVD Proxy obtains the integrity measurement m of the given compartment ID from the CompartmentManager. If the value m is listed in the TVD policy P as allowed to join, the TVD Proxy configures the underlying resource management to connect the compartment to the virtual resources of the TVD, e.g., "plugging" a virtual network connector to the VM.[8]

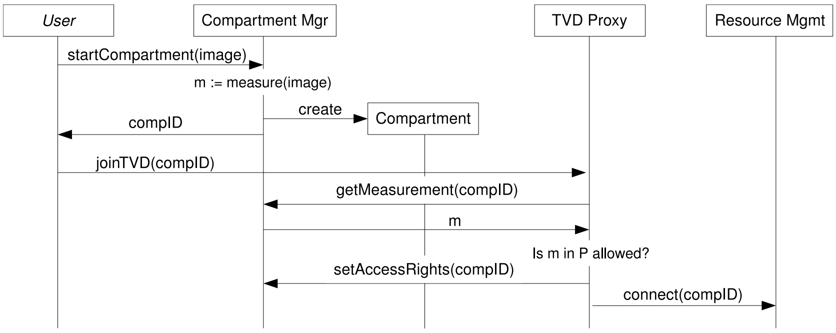

Fig. 6. TVD join protocol.

4 Remaining Challenges and Related Work

Privacy policy languages are designed to translate the privacy policies for users and organizations into statements that can be interpreted by IT systems. In [10] the authors give an overview of common policy languages. W3C's Platform for Privacy Preferences (P3P) was designed to express website privacy policies in machine-readable format [11], and P3P *Preference Exchange Language* (APPEL) is used to express privacy preferences of an individual and to query the P3P data[12,13]. *CPExchange* was developed to facilitate business-to-business communication about privacy policies [14]. For internal privacy policies of organizations, IBM proposed *Enterprise Privacy Authorization Language* (EPAL)

[8] The details of the resource isolation and realization of TVDs on this level are out of scope for this paper. Cabuk et al. [7] show how to realize network isolation based on VLAN tagging.

[15]. Another language for describing both privacy and security policies in a machine readable format is the *eXtensible Access Control Markup Language* (XACML) [16]. Other initiatives, such as DPAL [17], and XPref [18], addressed various aspects of expressing privacy requirements and related concepts. Due to the growth of services that require the transfer of context sensitive information (e.g., time and location), the Internet Engineering Task Force (IETF) initiative started work on *Geopriv*, a language that can express policies for granting access on the basis of presence and location information [17].

In addition to the earlier work on access control policies and (privacy) languages, recent research has analyzed and developed methodologies for evaluating actual policies to compare them with the policies the users desired to use, e.g., Bauer et al. [19] conducted user study of access control policies. Cornwell et al. [20] have analyzed policy management in different applications in mobile computing and developed applications where users can define policies to control the usage of private information, e.g., location-based or contextual information. Sadeh et al. [21] analyzed user interfaces for policy definition and mechanisms for auditing the disclosure of private information.

We conclude that, while the need to ensure user control and enforcement of privacy policies was recognized, most research so far focuses on formal languages defining privacy and related policies in various contexts, user requirements for such policies, and approaches for applications to incorporate user controlled flexible policies. However, little attention was given to the mechanisms to support automatic enforcement and interpretation of these policies. In this paper, we propose an approach to policy enforcement that takes into consideration the results of earlier research, including user requirements and design of formal policy languages. The new framework offers a realistic approach to the control and enforcement of privacy policies in a variety of contexts. We think that TVDs can help construct the *privacy domains* to support privacy protection of sensitive data that need to be shared. The process to build domains where the protection of sensitive data is governed by privacy policies determined by users still needs to be defined. Policy management for privacy domains remains a major challenge as complex privacy policies need to be enforced within a domain, when a machine joins or leaves the domain, and for inter-domain communication.

The idea of the Trusted Personal Information Wallet is derived from previous work [22], which uses a password wallet as authentication agent to access web sites. It protects private data (credentials) of a user during the authentication to a remote server. This approach uses Trusted Computing technology to ensure that the wallet is executed in a trusted environment. In addition to protecting the credentials, *SpyBlock* [23] protects against the unintentional disclosure of sensitive information (like credit card numbers, name, address, etc.) as a result of malicious transactions [24].

Since the Trusted Personal Information Wallet acts as an agent for the user's private data and it can migrate to other platforms, it is comparable to mobile agents. Wilhelm et al. [25] propose to use a tamper-resistant hardware to provide a secure execution environment for mobile agent code. Balfe and Gallery [26] outline how attestation can be used to ensure that an agent only visits host platforms behaving in an expected manner and that access to the private agent data complies to the desired security policies. In [27], the main approach is the protection of an agent's private cryptographic key by binding the key to a TPM. In contrast, the wallet (agent) in the framework proposed here does not directly use the TPM, but relies on the TVD infrastructure to (automatically) deploy a trusted execution environment and enforce privacy policies.

5 Conclusion

In this paper, we proposed a conceptual framework for privacy policy management and enforcement to ensure security and trust for sharing of private or sensitive information. We believe that Trusted Computing technology, in particular the concept of trusted virtual domains (TVDs), can efficiently support privacy policy enforcement. We think that future research will lead to the development of trusted privacy-enhancing architectures that will be applicable to several use cases, e.g., e-commerce, enterprise rights management, e-health, and other areas. Here we outline only the first steps towards the definition of such architectures. In addition, the definition and enforcement of more complex privacy policies will be a subject of future work.

Acknowledgment This work has been partially funded by the European Commission as part of the OpenTC project (`http://www.opentc.net`).

References

1. Anti Phishing Working Group: Phishing Activity Trends Report(s) (2005-2007) `http://www.antiphishing.org`.
2. Evers, J.: Phishers get personal (May 2005) `http://news.com.com/Phishers+get+personal/2100-7349_3-5720672.html`.
3. Sailer, R., Valdez, E., Jaeger, T., Perez, R., van Doorn, L., Griffin, J.L., Berger, S.: sHype: Secure hypervisor approach to trusted virtualized systems. Technical Report RC23511, IBM Research Division (February 2005)
4. Griffin, J.L., Jaeger, T., Perez, R., Sailer, R., van Doorn, L., Cáceres, R.: Trusted Virtual Domains: Toward secure distributed services. In: Proceedings of the 1st IEEE Workshop on Hot Topics in System Dependability (HotDep'05). (June 2005)
5. Bussani, A., Griffin, J.L., Jansen, B., Julisch, K., Karjoth, G., Maruyama, H., Nakamura, M., Perez, R., Schunter, M., Tanner, A., Doorn, L.V., Herreweghen, E.A.V., Waidner, M., Yoshihama, S.: Trusted Virtual Domains: Secure foundations for business and IT services. Technical Report RC23792, IBM Research (2005)
6. Trusted Computing Group: TPM main specification, version 1.2 rev. 103 (July 2007) `https://www.trustedcomputinggroup.org`.
7. Cabuk, S., Dalton, C.I., Ramasamy, H., Schunter, M.: Towards automated provisioning of secure virtualized networks. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07), ACM Press (2007) 235–245
8. Berger, S., Cáceres, R., Pendarakis, D., Sailer, R., Valdez, E., Perez, R., Schildhauer, W., Srinivasan, D.: TVDc: Managing security in the trusted virtual datacenter. SIGOPS Oper. Syst. Rev. **42**(1) (2008) 40–47

9. Katsuno, Y., Kudo, M., Perez, P., Sailer, R.: Towards Multi-Layer Trusted Virtual Domains. In: The 2nd Workshop on Advances in Trusted Computing (WATC'06 Fall), Tokyo, Japan, Japanese Ministry of Economy, Trade and Industry (METI) (November 2006)
10. Kumaraguru, P., Cranor, L., Lobo, J., Calo, S.: A survey of privacy policy languages. In: Workshop on Usable IT Security Management (USM 07): Proceedings of the 3rd Symposium on Usable Privacy and Security, ACM (2007)
11. Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J.: The Platform for Privacy Preferences 1.0 (P3P 1.0) specification. Technical report (April 2002)
12. Cranor, L.: Web Privacy with P3P. O'Reilly & Associates (September 2002)
13. Cranor, L., Langheinrich, M., Marchiori, M.: A P3P Preference Exchange Language 1.0 (APPEL 1.0). Technical report (June 2005) WWW Consortium.
14. Bohrer, K., Holland, B.: Customer Profile Exchange (CPExchange) Specification, Version 1.0. Technical report (October 2000)
15. Schunter, M., Ashley, P., Hada, S., Karjoth, G., Powers, C.: Enterprise Privacy Authorization Language (EPAL 1.1). Technical report, IBM (2003)
16. Moses., T.: eXtensible Access Control Markup Language (XACML) version 2.0. Technical report, Oasis (2005)
17. Schulzrinne, H., Tschofenig, H., Morris, J., Cuellar, J., Polk, J., Rosenberg, J.: A document format for expressing privacy preferences. `http://tools.ietf.org/html/draft-ietf-geopriv-common-policy-11` (August 2006)
18. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: An XPath-based preference language for P3P. In: WWW'03: The 12th International Conference on World Wide Web. (2003) 629–639
19. Bauer, L., Cranor, L.F., Reeder, R.W., Reiter, M.K., Vaniea, K.: A user study of policy creation in a flexible access-control system. In: SIGCHI Conference on Human Factors in Computing Systems (CHI'08), ACM (2008)
20. Cornwell, J., Fette, I., Hsieh, G., Prabaker, M., Rao, J., Tang, K., Vaniea, K., Bauer, L., Cranor, L., Hong, J., McLaren, B., Reiter, M., Sadeh, N.: User-controllable security and privacy for pervasive computing. In: 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile 2007), IEEE (2007)
21. Sadeh, N., Hong, J., Cranor, L., Fette, I., Kelley, P., Prabaker, M., Rao, J.: Understanding and capturing people's privacy policies in a mobile social networking application. Journal of Personal and Ubiquitous Computing (2008)
22. Gajek, S., Sadeghi, A.R., Stüble, C., Winandy, M.: Compartmented security for browsers – or how to thwart a phisher with trusted computing. In: 2nd Intl. Conference on Availability, Reliability and Security (ARES 2007). (2007) 120–127
23. Jackson, C., Boneh, D., Mitchell, J.: Spyware resistant web authentication using virtual machines. `http://crypto.stanford.edu/spyblock/` (2006)
24. Jackson, C., Boneh, D., Mitchell, J.: Transaction generators: Root kits for web. In: 2nd USENIX Workshop on Hot Topics in Security (HotSec '07). (2007)
25. Wilhelm, U.G., Staamann, S.M., Buttyan, L.: A pessimistic approach to trust in mobile agent platforms. IEEE Internet Computing **4**(05) (2000) 40–48
26. Balfe, S., Gallery, E.: Mobile Agents and the Deus Ex Machina: Protecting Agents using Trusted Computing. In: Proceedings of the 2007 IEEE International Symposium on Ubisafe Computing (UbiSafe-07), IEEE Computer Society Press (2007)
27. Xian, H., Feng, D.: Protecting mobile agents' data using trusted computing technology. Journal of Communication and Computer **4**(3) (2007) 44–51