A Hybrid Approach for Private Data Protection in the Cloud

Amal Ghorbel[1(✉)], Mahmoud Ghorbel[1(✉)], and Mohamed Jmaiel[1,2(✉)]

[1] ReDCAD Laboratory, National School of Engineers of Sfax, University of Sfax, B.P. 1173, 3038 Sfax, Tunisia
{amal.ghorbel,mahmoud.ghorbel,mohamed.jmaiel}@redcad.org
[2] Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, 3021 Sfax, Tunisia

Abstract. With the emergence of the cloud computing paradigm, the personal data usage has raised several privacy concerns like the lack of user control, the non-compliance with the user's preferences and/or regulations, the difficulty of the data flow tracking, etc. In particular, one unsolved problem is to ensure that customers data usage policies are enforced, regardless of who accesses the data, how they are processed, where are the data stored, transferred and duplicated. This issue calls for two requirements to be satisfied. First, data should be handled in accordance with both owners' preferences and regulations policies whenever it exists in the cloud and throughout its lifetime. Second, a consistent data flow tracking should be maintained to follow up the data derivation. Toward addressing these issues, we propose in this paper a hybrid approach to protect private data in the cloud. We propose the PriArmor data content for self-defending data when stored or transferred in the cloud and the PriArmor agent that acts as an armor for its privacy protection when processed by the cloud-based services. To facilitate the policy specification, we propose a novel privacy ontology model that drives the data owner to express his privacy requirements and to consider the regulations policies. Finally, we present the implementation details as well as a demonstration that shows flexibility and efficiency of our approach.

Keywords: Privacy · Cloud · Privacy policies · Ontology · Policy enforcement

1 Introduction

There is no doubt that cloud computing presents a real evolution in the IT world that offers many advantages for both particular and professional customers. In fact, reduction of IT costs and improvement of business agility are identified as the greatest assets for the cloud adoption. Nevertheless, the lack of trust and transparency about data handling, is at present, a key inhibitor in moving to the public cloud [1]. For cloud customers, there are many ambiguities regarding the data outsourcing in such remote hosting paradigm where data are stored and processed in remote machines not owned or controlled by the data owners [2].

M.H.A. Au et al. (Eds.): GPC 2017, LNCS 10232, pp. 23–37, 2017.
DOI: 10.1007/978-3-319-57186-7_3

Therefore, the data owners don't have a clear idea about how and by whom their data are accessed, stored and used. Even worse, due to replication and backup cloud mechanisms, many copies of data may be created and stored on servers in different geographical locations where local laws can bring more privacy risks.

One major privacy concern in the public cloud is the data access and usage policy enforcement. This concern can cover the overall privacy issues as all aforementioned problems can be regulated by expressing well-defined privacy policies and enforcing them. Nevertheless, the majority of the current customers are unaware of the risks that threaten their private data in the public cloud environment, and thus, they are not able to define the appropriate policies that assure the protection of their data. Furthermore, the privacy policies are various and complex since they encode different access and usage restrictions including customer's business and regulatory requirements. A well-known regulatory requirement for data privacy is the physical location of the data (e.g., European regulation prohibits to store or transfer medical information out of the European fence [3]). Thus, it is not trivial for the cloud customer to express policies that exactly meets privacy requirements (customer's and regulation requirement) and at the same time considers all risk of data usage in the cloud. Second, ensuring that these policies are respected, is at present an important challenge. A commonly adopted approach is to associate policies with data while moving across the cloud system. Hence, data usage is permitted unless the attached policies are respected. To do this, some research works propose to emerge high-level mechanism to ensure that privacy policies are enforced before accessing the data. The main shortcoming of such solutions is the loss of control over the data once the usage request is checked and the data is released to the data actor. In fact, these approaches assume that the authorized users are trusted not to illegally leak or disseminate the data. Hence, they cannot prevent intentionally or inadvertently data misuse of the authorized users like keeping copies of data, sending data to unauthorized third parties, etc. Another policy enforcement concern involves the fact that outsourced data are always moving across the cloud ecosystem due to its dynamic feature. They can be stored, transferred and duplicated in forbidden locations specified by the data owner policy or regulation policies. Such actions are performed by the cloud service provider (CSP) using cloud infrastructure means (e.g., cloud manager, cloud balancer, etc.) and generally do not require permission to be performed on the data. Hence, privacy policy concerning data location can be easily and implicitly violated. The challenge here is that the integration of enforcement mechanisms in the cloud infrastructures in not a viable option.

In this paper, we propose an integrated approach for private data protection that enables cloud customer and regulation policy specification and that introduces new privacy structures and mechanisms for ensuring policy enforcement. We introduce the PriArmor (Privacy Armor) agent that enforces privacy policy on detected explicit data handle by cloud-based services (e.g., copy, share, collection, etc.). We introduce the new structure of data content named PriArmor data for self-location checking to deal with CSP implicit handling of data

that cannot be detected by the enforcement mechanisms (e.g., transfer, storage, duplication). To enable policy specification, we propose a novel privacy ontology model as a semantic way to express access and usage control constraints. Based on this model, we introduce the Privacy Ontology-based Framework (POF) to be used by the data owner for automating the privacy policy specification. The POF integrates predefined domain specific ontologies that enable to consider regulatory policies. After the policies specification, the POF builds the PriArmor Data. The PriArmor data can be only used by a PriArmor agent VM which is integrated into the cloud infrastructure. This agent acts as an armor for data when it is processed by untrusted cloud-based services (application launched on the PriArmor agent VM). The PriArmor agent contributes by tackling the related problems: (1) providing appropriate guarantees to enforce policies contained in the PriArmor data, (2) tracking the protected data flows across the cloud environment and making clones of PriArmor data to include the derived data.

Problem statement. This paper address the problem of privacy requirement including customer's preferences and regulation policies regarding data access and usage in the cloud environment. The data access and usage control must be enforced on implicit and explicit actions done in the cloud. To this end, we study the three main problems below:

1. The data owner is unaware of data usage in the cloud and thus, he is not able to define the appropriate policies that assure his data protection. Moreover, it is not trivial for a data owner to explore regulation laws and to specify them as machine readable policies.
2. Actual enforcement mechanisms do not consider all cloud aspects in the enforcement process (dynamic data management, data duplication and transferring, etc.). Moreover, they do not consider implicit CSP data handling which leads generally to location-related policy breaches.
3. An enforcement mechanism must ensure a low-level usage control enforcement. Such mechanism requires knowing data flow to protect the derived data.

Contribution. We propose a hybrid approach that banks on several privacy solutions and contributes by tackling the three related sub-problems:

1. Enabling data owner preferences definition and legal policies consideration.
2. Providing appropriate guarantees to enforce location-related policy for the CSP implicit data usage without impacting cloud infrastructures.
3. Ensuring system call level policy enforcement for cloud-based services for explicit data handling and enabling data flows tracking across the system to protect the derived data.

The rest of this paper is structured as follows. In Sect. 2, we introduce a motivating use case. In Sect. 3, we present and detail the proposed approach. Section 4 introduces the data encryption and the key management solution. Section 5 includes some implementation details and a demonstration of the proposed approach. In Sect. 6, we analyze some related works. Section 7 concludes the paper.

2 Motivating Use Case

In order to have a clear view of the issues previously described, we have identified one critical scenario of privacy threats in cloud environments that will be used as a reference example in this paper.

We consider an insurance company serving both private customers and companies. In order to improve its business and to increase elasticity, the insurance decides to adopt cloud-based services for email, customer data storage, collaboration, and website. Consider a customer, that resides in Europe, who want to ask for health insurance offers using the insurance website. To have a personalized offer, the insurance asks the customer to provide some private information such as name, address, age, sex, job, an overview of his history of diseases and his Electronic Health Record (EHR), etc. Before releasing such information, the customer must have a way to define his privacy settings by specifying which entity can access his data, for what purposes, where it can be stored, what are the obligations that must be satisfied before or after the access to the data (e.g. notification, retention). He must also have a way to consider regulation policies and to integrate them with his privacy preferences. The insurance also must ensure that the used cloud-based services integrate an enforcement mechanism that enables compliance with data owner preferences and the corresponding regulation policies.

To allay customer' concerns, it is essential to provide an effective mechanism for users to specify their privacy preferences and to ensure their protection in the cloud. In our model attack, we try to protect the data when it is processed or stored in the cloud. Hence, the following threats are out of the scope of this paper: (1) taking a photo of the screen, (2) human memory, (3) peeking in the VM that deploys the cloud-based service when running and making copies of it.

3 The Proposed Approach

The main contribution of this paper is to provide an integrated approach to protect private data in the cloud throughout its lifetime (at rest, in use and in-transit). To enable this, we propose the privacy policy ontology (Sect. 3.1) and the Privacy Ontology-based Framework (Sect. 3.2) for privacy requirements specification. We introduce the PriArmor data container (Sect. 3.3) and the PriArmor agent (Sect. 3.4) to ensure the policy enforcement in the cloud ecosystem. We refer to the next subsections for an in-depth description.

3.1 Privacy Ontology Model

Policy specification is a very important step to ensure the data protection in the cloud. The data owner must specify a set of policies to express his preferences concerning the data access and usage. One major issue here is that most of the cloud customers don't have a clear idea about practices that data actors can perform with their private data. An unstudied specification of policies may create

vulnerabilities which can be exploited to data misuse or leakage (for example a customer who is not aware of data collection and sharing in the cloud may not specify policies to regulate these practices). Further, the diversity and the complexity of the privacy policies, expressed either by legal requirement or by the data owner preferences, explain the need for a semantic modeling of the privacy policies to abstract its complexity and facilitate its automation and enforcement. To this end, we propose the privacy ontology model that catches and gathers various concepts related to the data access and usage in the cloud. Our privacy ontology model is split into two ontologies: the policy ontology and the data usage ontology. The two ontologies are represented respectively by Figs. 1 and 2. A description of their ontological concepts and relations are provided as follows.

Policy Ontology. The policy ontology represents the privacy policy and all its properties.

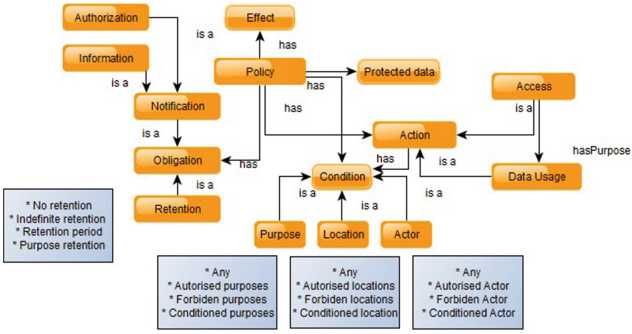

Fig. 1. Ontological concepts of privacy policy

- Protected data: a policy is defined for a target private data.
- Action: data action can be an access or a data usage. An action has an actor which is the data requester. The data owner can specify authorized, forbidden, or conditioned (that demand notification of data owner for information or for authorization) actors. Data usage presents the access purpose (collection, share, copy, display, etc.). The data owner can identify authorized, forbidden, or conditioned access purposes.
- Obligation: the obligation concept concerns retention and notification. For data retention, the data owner can specify indefinite retention when the period of retention is undefined, period retention to identify time period of retention of the collected data or purpose retention when data are retained only during the time period necessary to complete the purposes. A notification can be informational (to inform the data owner about an event) or seeking for authorization actions.
- Condition: each action has a condition that is defined by a set of context variables such as actor, the purpose of use, usage location, recipient of private data, the location of the recipient. If the condition evaluates to true based

on the values of the context variables, then the action is permitted or denied according to the policy effect.
– Effect: can be either allow or deny.

Data action ontology. In the data action ontology, we enumerate the possible data processing in the cloud and the corresponding conditions and obligations.

– Collection and Share: each of these concepts have a purpose and a recipient that has a location and (e.g., maintenance, marketing, statistical, personalization, profiling, etc.). The data owner must identify authorized, forbidden, or conditioned recipients, location recipient, and purposes.
– Transfer, Storage, and Duplication: these actions are generally performed implicitly by the CSP. We require that such actions must be done only on encrypted data.
– Read, Modification, Copy, and Deletion: each of these actions has a purpose and usage location. Data owner must identify authorized, forbidden or conditioned purposes and location.

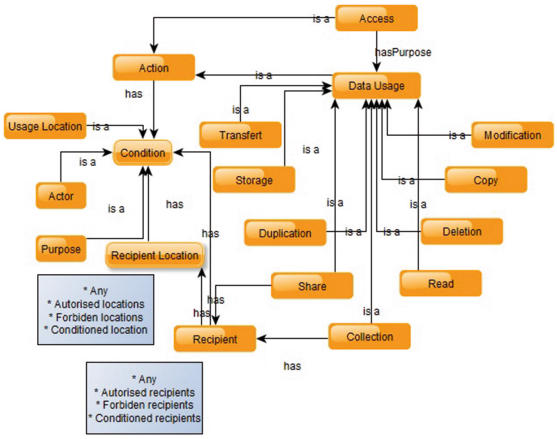

Fig. 2. Ontological concepts of data usage

3.2 Privacy Ontology-based Framework

We propose the Privacy Ontology-based Framework (POF) that implements the proposed ontology model to facilitate the policy generation. Our framework provides a graphical tool that hides the complexity of the ontology representation from the users. An ontology transformer, a consistency auditor and an efficiency tester are integrated to generate readable and coherent policies. The overall process of the policy generation is depicted in the Fig. 3.

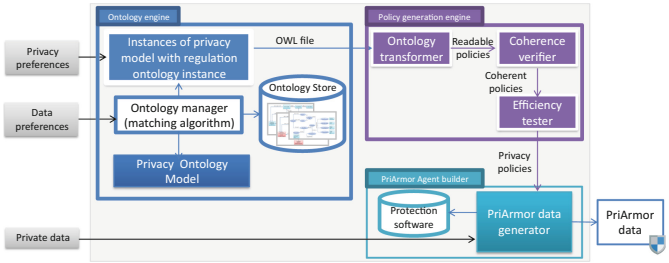

Fig. 3. Privacy Ontology-based Framework

At first, it is necessary for the data owner to identify the domain of the data (e.g., medical data, financial data, personal data, etc.), his location and many others preferences. An ontology matching algorithm is defined for the selection and the instantiation of the appropriate regulatory policy based on the specified preferences (see Algorithm 1) to be associated with our privacy ontology model. The data preferences (d: domain, l: location) are the inputs of the algorithm, *or* is the selected regulation ontology according to the data preferences and *orIns* is the instantiation of the selected ontology which is the output of the algorithm. Now, the data owner can express his privacy preferences concerning the data access, the data usage and the associated obligations to complete the ontology model specification using the provided graphical editor. The output from the graphical tool is an OWL file that contains all needed features (privacy regulatory and data owner preferences) to generate the policies. From the resulted OWL file, the ontology transformer automatically generates readable policies in a specific language (e.g., XACML [4], PPL [5], KAoS [6], etc.). Thereafter, the consistency auditor verifies the coherence of the policies and then the efficiency tester checks whether the generated policies gratify the data owner expectations or not. This last module avoids the user to have a syntactically correct policy but with unexpected effect. Finally, the POF encapsulates data with the policies and associates them with the protection software to build the PriArmor data. The integrated protection software is used to enforce the location-related policy using self-location checking and self-destruction algorithms. The PriArmor data also integrates a self-integrity checking software to enables it to protect itself. Now, the produced data content is ready to be uploaded to the cloud.

3.3 PriArmor Data

The PriArmor data is a movable container that encapsulates sensitive data and privacy policy and assures its location-related policy enforcement through the provision of effective software packages as shown in Fig. 4. These packages are signed and sealed by a known Trust Authority (TA). Based on a self-destruction mechanism, the introduced software packages enable:

Self-integrity checking. The PriArmor data checks the integrity of its contents at any random time. It computes the hash value for the sensitive data, the

Algorithm 1. Ontology Matching Algorithm

```
procedure RegOntoIns
    Input = data preferences;
    \\(d: domain, l: location, etc.)
    Variable or : ontology model; orIns : instantiated ontology;
    Request all regulation ontology from ontology store;
    for each predefined regulation ontology do
        if ontology regulation matches input data preferences then
            or = selected ontology
            orIns = instantiate (or)
            Return orIns
        else
            Return "Data has no regulation law";
        end if
    end for
end procedure
```

policies, and the protection software. Then, it verifies the signed hash value by comparing it to the computed hash value. If the verification fails, the PriArmor data performs the self-destruction mechanism to prevent compromises.

Self-location checking. The data cloud actor, such as the CSP, may simply store, transfer, or duplicate the PriArmor data without trying to access the data and to verify the associated policy. Such actions can be not permitted if the destination does not match the location requirements. Hence, the PriArmor data performs self-location checking at any random time and before being handled by any PriArmor agent. The PriArmor data performs a self-destruction if its physical location is prohibited by the defined policies.

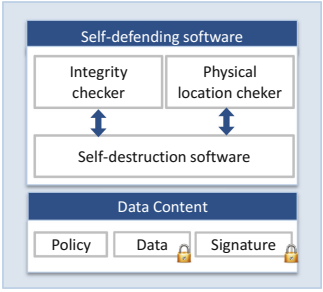

Fig. 4. PriArmor data content

3.4 PriArmor Agent

The PriArmor data structure can be only decrypted and handled in a PriArmor agent VM that is integrated into the cloud infrastructure. The PriArmor agent

enables low level enforcement of the privacy policies using a System Call Interception (SCI) technique. By using such technique, the PriArmor agent is able to detect and prevent the intention for the policies breaching. Again, the PriArmor agent incorporates a data flow model that tracks private data and reflects the existence of the derived data through the system Fig. 5. The main functionalities of the PriArmor agent are:

Authentication. The PriArmor agent includes a claim-based authentication that requires a security token issued by the Trusted Authority (TA). TA authenticated the user at an early stage through a set of claims that specify the user identity, roles, assigned permissions, location etc. The authentication is done through the PriArmor agent interface whenever a user or an application (cloud actor) requires the usage of the protected data.

Active control. The PriArmor agent implements a data flow model that allow to monitor all actions performed on the data and detects the derived data (copies of data, modified data, appended data, etc.) from the data usage.

Policy enforcement. After authentication of authorized users, the PriArmor agent intercepts all requests for data usage, checks for their compliance with the defined policy and decides whether to allow or to deny the access to the protected data. The PriArmor agent is an instantiation of a generic representation of the policy management components which implement a system call interception solution. It consists of three components: the request analyzer, in charge of intercepting system calls and enforcing the corresponding privacy policy and obligations; the evaluation engine, responsible for deciding about the admission of the intercepted system call; and the metadata store, implements a data flow model and maintains a connection between data and its derivation within the system.

PriArmor engine. The PriArmor engine enables the PriArmor data generation whenever a data or its derivation must leave the active PriArmor VM. This engine clones the original PriArmor data to include the derived data.

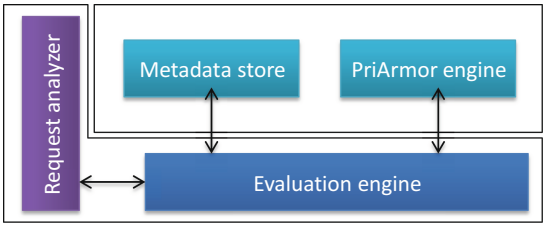

Fig. 5. PriArmor agent architecture

4 Data Encryption and Key Management

The PriArmor solution includes a symmetric-key cryptographic scheme to protect the larger-size content (protected data) and an asymmetric cryptographic scheme for protecting the smaller-size items (the symmetric encryption key Ku_{sym}).

Creation and encryption: we describe the implementation of the key management in our solution. The format of the protected data generated from the POF is shown in Fig. 6. To create the PriArmor data, the owner first generates a new symmetric key (Ku_{sym}), and uses it to encrypt the data. (Ku_{sym}) is then encrypted using the PriArmor agent public key (Ka_{pub}). The POF associates the protection software to builds the PriArmor data. Then, it calculates a cryptographic hash over the encrypted Ku_{sym}, the encrypted data, the policy and the included software. The hash is signed by the TA using its private key (Kta_{pri}) to ensure the data content control.

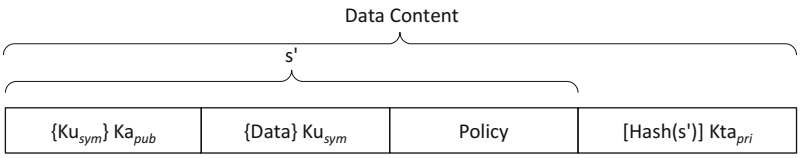

Fig. 6. Format of the protected data content

Transfer to the cloud: once the PriArmor data is created, it can be moved to the cloud. The PriArmor data is a self-location checking content, hence it can be stored and processed anywhere in the system.

Self-integrity checking: at any time, the PriArmor data checks for his integrity. It calculates the hash of the encrypted Ku_{sym}, the encrypted data, the policy and the protection software. Then, it compares it with the decrypted hash in the signature to ensure that they have not tampered.

Decryption: the PriArmor data can only be processed by a PriArmor agentVM. The latter follows a secure protocol to decrypt the protected data. First of all, it ensures that the requesting user is authenticated and that the user owns a security token issued from a known TA, otherwise the access is denied. Then, the PriArmor agent retrieves the policy and compares it with the request. If the decision is to grant access to the data, the PriArmor agent uses its private key Ka_{pri} to decrypt the Ku_{sym} key and then uses Ku_{sym} to decrypt the data. The Ka_{pri} key is stored in a TPM (Trusted Platform Module) tamper-resistant hardware component that provides a shielded location to protect secret keys [8].

Re-encryption: if an authorized data processing requires the data to leave the active PriArmor agent VM (send, share, etc.), then the PriArmor engine searches for the originator PriArmor data in the metadata store. It encrypts the

data using Ku_{sym}. Then, if the data will be sent to another Ku_{sym} agent VM, it encrypts the Ku_{sym} using the destination public key. Otherwise, the Ku_{sym} is encrypted using Ka_{pub}. The PriArmor engine integrates a signed hash over the encrypted Ku_{sym}, the encrypted data, the policy and the protection software in the data content and builds the PriArmor data.

5 Implementation Details

5.1 Prototype

Our approach introduces the POF that enables cloud customer's and regulation policy specification. It also includes a protection mechanism that is based on the PriArmor solution. The principle of our mechanism is as follows: the PriArmor data can be only processed in a PriArmor agent VM that we assume its existence before the PriArmor data generation.

The POF facilitate the data owner mission for the policy specification by introducing a graphical tool and based on the proposed privacy ontology model. We carry on exploiting OWL [9] description language to implement privacy ontology model. First of all, the POF requires the data preferences that are the inputs of the regulation ontology instantiation algorithm. Next, an instance of the privacy ontology model is loaded including the selected regulation ontology instance. Thereafter, the POF guides the data owner to fill in the rest of the ontological concepts of our model by providing him a set of interfaces for specifying each data action. Second, the POF transforms the instantiated ontology into privacy policy (the current implementation of the POF supports the XACML policy). At present, the implementation of the POF does not support the consistency auditor and the efficiency tester. We aim to integrate it in the future work. Once the policies are generated, the POF bounds the data to the policies as described in Sect. 4 and associates them with the protection software packages to build the PriArmor data.

The software packages of the PriArmor data implementation banks on Vanish[1] core system [10]. The central security goal of Vanish software is to ensure the destruction of data regardless of whether it is copied, transmitted, or stored in a distributed system. Hence, the data becomes unreadable after a predefined period. We modify the Vanish data destruction software in a way that it will be triggered by the self-integrity checking and self-location checking software.

We implement a PriArmor agent prototype on an OpenBSD VM. Systrace framework [11], which is integrated by default in the OpenBSD, has been used to implement the request analyzer and the evaluation engine components. Using Systrace, no modifications to the operating system itself are needed. The metadata store is implemented as a software layer on the top of the systrace device */dev/systrace*. This software keeps track of protected data using the generic data flow model introduced in [7]. This model allows fine-grained data flow tracking at the system call level. The PriArmor engine software is launched whenever data

[1] https://vanish.cs.washington.edu/index.html.

are allowed to leave the actual PriArmor agent VM. Hence, PriArmor engine communicates with the metadata store to identify the original PriArmor data and clone it for the derived data.

5.2 Demonstration

To show the usefulness of our approach, we present a demonstration of the proposed mechanisms. We rely on additional assumptions in our demonstration that are: (i) the attacker does not have administrator privileges; (ii) the underlying OpenBSD operating system is free of vulnerabilities. Considering the use case presented in Sect. 2, we suppose that the insurance company has launched her services on a PriArmor agent VM located in Italy and has used cloud-based storage services located in Germany (see Fig. 7). Since Bob (the cloud customer) lives in France and that he will deliver his EHR then the POF will select and instantiate the predefined ontology of the European regulation for health data. The European directive requires that EHR must never leave the European fence. Hence, a policy that inhibits data transfer out on the European fence is defined. Now, the PriArmor data is uploaded to the PriArmor agent VM. The company's employee is authenticated and is permitted to view the sensitive data but not to disseminate it. Assume that the employee tries to share data with another unauthorized party. These practices are inhibited based on the PriArmor agent enforcement mechanism. Assume that Bob accepts the offer proposed by the company. His EHR is re-packaged in PriArmor data container (using the PriArmor engine) and is stored in the storage server S1 (Germany). Suppose that the S1 cloud manager performs duplication of Bob's PriArmor data and transfer it to be stored in the storage server S2 (US) to ensure the data availability. At any random time and before being used, the PriArmor data performs self-location checking and note that his location in forbidden by the specified policy. Hence, it runs the self-destruction algorithm.

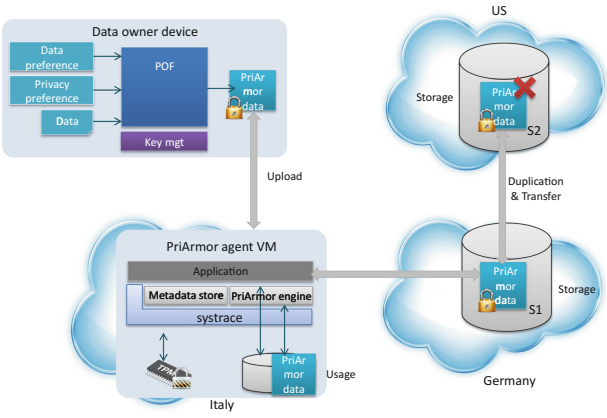

Fig. 7. Scenario of the proposed approach

6 Related Works

Various technologies have been introduced to deal with privacy compliance issues in the cloud. In the following, we give an overview of some related works. The first category of approaches focuses on how to protect data from the user side. The user is involved either in the expression of his preferences or in the consideration of legal texts. Rahmouni proposes in [12] to semantically model privacy obligations of legal, ethical or cultural nature. She aims to formalize privacy policies intercepted from the EU directive through the use of ontology modeling (OWL) and semantic web rule language (SWRL). These policies are then mapped to the XACML language in order to be enforced in the cloud. Nevertheless, this approach considers only legal policies (does not consider data owner preferences) and only the EU directive ones. Similarly, Papanikolaou et al. come up with a toolkit for automating compliance in cloud computing services [13]. This toolkit allows the semantic annotation and natural-language processing of policy texts (regulation text and/or data owner preferences text) to generate machine-readable rules. However, these user-centric approaches remain incomplete since they are not able to ensure enforcement of generated policies in the cloud.

Other solutions introduce data-centric approaches that emphasize mechanisms and techniques to automate sensitive data protection anywhere in the cloud. Squicciarini et al. introduce the Self-Controlling Objects (SCO) [14]. This object that encapsulates sensitive data along with their policies and assures their protection by means of object-oriented programming techniques. Each time, the access to the SCO protected content is attempted, its policy is evaluated according to the requester's credential and location. The SCO manages copies of data synchronizes and updates it if there was a change. Angin et al. introduce an entity-centric approach for identity management in the cloud [15]. This entity incorporates the data, the policy and a VM that manages the policy enforcement. This entity can perform a set of protection mechanisms to protect themselves such as integrity checks, apoptosis, evaporation and decoy. However, these approaches can only enforce access policy and loss the control over the data once access is granted.

The approach presented in [16,17] proposes to introduce a data protection module named CDPM (Client Data Protection Module) deployed in the user side to allow policies generation. To ensure policies enforcement in the cloud, authors propose to integrate the File Data Protection Module(FDPM) within the VM system file. FDPM will intercept, control and trace all operations done by applications on sensitive files. Castiglione et al. [18] propose an engine for lossless dynamic and adaptive compression of 3D medical images, which also allows the embedding of security watermarks within them. Gosman et al. [19] illustrate a security model for Intelligent Transportation Systems (ITS) where participants can specify how data sharing captured by an ITS application will behave in regards to their own privacy requirements. The proposed solution is able to maps the differences between ITS applications requirements regarding data usage and the user privacy constraints related to the location and timestamp of his shared data.

All the works outlined above are interesting. However, each of them has one of these drawbacks. First, we can remark that the majority of these approaches do not consider privacy policy specification or assume that data owner is able to express his policy. Second, part of the presented approaches consider only access control enforcement and loss the control once data are delivered. Besides, the majority of works cannot track the flow of data to maintain the same level of enforcement. Moreover, they do not consider the dynamic feature of cloud that probably leads to location-related policies violation.

7 Conclusion

In this paper, we propose a new policy based technology for privacy preservation in the cloud. We believe that the privacy protection in public cloud is a complex process that requires the implication of all involved entities to ensure the data protection throughout its lifetime. Thereby, in our solution, we have adopted a hybrid approach that involves the data, the data owner and the data actors in the data protection process. In fact, we propose a novel privacy ontology model that catches various concepts related to the data access and the usage in the cloud and considers multiple aspects of this environment. We propose the Privacy Ontology-based Framework (POF) that drives the data owner for specifying his privacy preferences based on the proposed privacy ontology model. Our framework includes predefined regulatory policy ontologies to consider privacy laws. The POF generates a new data content structure named PriArmor data that bundles the sensitive data, the defined policy and a set of self-defending software. These software include self-integrity and self-location checking algorithms that are able to launch a self-destruction k whenever there was a location-related policy or PriArmor data content violation. PriArmor data can be processed in PriArmor agent VM that ensures the policy enforcement and the data flow tracking at system call level. We present the data encryption and key management used to generate the PriArmor Data and provide implementation details about the prototype architecture. We demonstrate the effectiveness of our approach and that is a viable option in real-life examples.

References

1. Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U.: The rise of "big data" on cloud computing: review and open research issues. Inf. Syst. **47**, 98–115 (2015)
2. Ghorbel, A., Ghorbel, M., Jmaiel, M.: Privacy in cloud computing environments: a survey and research challenges. J. Supercomput. 1–38 (2017)
3. EU Directive: 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Offcial J. EC **23**(6) (1995)
4. Moses, T.: Extensible access control markup language (XACML) version 2.0. Oasis Standard, 200502 (2005)

5. Trabelsi, S., Njeh, A., Bussard, L., Neven, G.: PPL engine: a symmetric architecture for privacy policy handling. In: W3C Workshop on Privacy and data usage control, vol. 4, no. 5, October 2010
6. Uszok, A., Bradshaw, J., Jeffers, R., Suri, N., Hayes, P., Breedy, M., Bunch, L., Johnson, M., Kulkarni, S., Lott, J.: KAoS policy and domain services: toward a description-logic approach to policy representation, deconfliction, and enforcement. In: Proceedings of the IEEE 4th International Workshop on Policies for Distributed Systems and Networks, POLICY 2003, pp. 93–96. IEEE, June 2003
7. Harvan, M., Pretschner, A.: State-based usage control enforcement with data flow tracking using system call interposition. In: Third International Conference on Network and System Security, NSS 2009, pp. 373–380. IEEE, October 2009
8. Chen, L., Mitchell, C.J., Martin, A. (eds.): Trust 2009. LNCS, vol. 5471. Springer, Heidelberg (2009)
9. McGuinness, D.L., Van Harmelen, F.: OWL web ontology language overview. W3C recommendation, vol. 10, 10 February 2004
10. Geambasu, R., Kohno, T., Levy, A.A., Levy, H.M.: Vanish: increasing data privacy with self-destructing data. In: USENIX Security Symposium, pp. 299–316, August 2009
11. Provos, N.: Improving host security with system call policies. In: Usenix Security, vol. 3, p. 19, August 2003
12. Rahmouni, H.B.: Ontology based privacy compliance for health data disclosure in Europe. Doctoral dissertation, University of the West of England, Bristol (2011)
13. Papanikolaou, N., Pearson, S., Mont, M.C., Ko, R.K.: A toolkit for automating compliance in cloud computing services. Int. J. Cloud Comput. 2 **3**(1), 45–68 (2014)
14. Squicciarini, A.C., Petracca, G., Bertino, E.: Adaptive data protection in distributed systems. In: Proceedings of the Third ACM Conference on Data and Application security and privacy pp. 365–376. ACM, February 2013
15. Angin, P., Bhargava, B., Ranchal, R., Singh, N., Linderman, M., Othmane, L.B., Lilien, L.: An entity-centric approach for privacy and identity management in cloud computing. In: 2010 29th IEEE Symposium on Reliable Distributed Systems, pp. 177–183. IEEE, October 2010
16. Betgé-Brezetz, S., Kamga, G.B., Dupont, M.P., Guesmi, A.: Privacy control in cloud VM file systems. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom), vol. 2, pp. 276–280. IEEE, December 2013
17. Betgé-Brezetz, S., Kamga, G.B., Ghorbel, M., Dupont, M.P.: Privacy control in the cloud based on multilevel policy enforcement. In: 2012 IEEE 1st International Conference on Cloud Networking (CLOUDNET), pp. 167–169. IEEE, November 2012
18. Castiglione, A., Pizzolante, R., De Santis, A., Carpentieri, B., Castiglione, A., Palmieri, F.: Cloud-based adaptive compression and secure management services for 3D healthcare data. Future Gener. Comput. Syst. **43**, 120–134 (2015)
19. Gosman, C., Cornea, T., Dobre, C., Pop, F., Castiglione, A.: Controlling and filtering users data in intelligent transportation system. Future Gener. Comput. Syst. (2016)