# ON THE SECURITY OF PUBLIC CLOUD STORAGE

Steffen Müller[1], Frank Pallas[2], and Silvia Balaban[3]

[1] st.mueller@kit.edu, [3] silvia.balaban@kit.edu
Karlsruhe Institute of Technology (KIT), Competence Center for Applied Security (KASTEL), Am Fasanengarten 5, 76131 Karlsruhe (Germany)

[2] frank.pallas@kit.edu
Karlsruhe Institute of Technology (KIT), Center for Applied Legal Studies, Vincenz-Prießnitz-Str. 3, 76131 Karlsruhe (Germany)

## Abstract

The broadly accepted and undisputed economic benefits notwithstanding, cloud computing, and particularly cloud storage, raises many security-related and legal qualms: Every enterprise considering to utilize cloud storage has to deal with compliance and security restrictions. In order to address these, cloud providers offer more and more security mechanisms for their services. At closer inspection, however, such mechanisms often are of limited value. In order to assess the security of existing cloud storage services, we build a generic usage model for public cloud storage integrating perspectives from the law and from economic agency theory as well as a respective basic threat model. Using these models, we then examine selected security mechanisms of two well-known public cloud storage services—Amazon S3 and Google Cloud Storage—and briefly sketch auspicious starting points for future research.

Keywords: Cloud Storage, Security, Threat Model, Usage Model, Data Protection, Agency.

## 1 INTRODUCTION

The major downside of—for other reasons often—advantageous cloud computing, and particularly of cloud storage, is that there are many security-related and legal qualms. In order to allow their customers to address and mitigate these qualms, cloud providers offer more and more security mechanisms for their services. For example, Amazon provides different ways of data encryption for Amazon Simple Storage Service (S3). Based on these mechanisms, application developers shall be able to securely employ cloud storage within higher-level applications.

The practical possibilities and implications of these mechanisms are, however, not yet sufficiently understood on a well-structured, sustainable, and transferable basis. With regard to the engineering of legally compliant cloud applications and for ensuring appropriate security within such applications in general, this is clearly unsatisfactory. In order to provide a first basis for analyses, assessments, and comparisons of existing cloud storage services in matters of security, we thus herein propose a generic usage model (section 3) as well as a basic threat model (section 4) for public cloud storage. Furthermore, we exemplarily apply these models to a single security threat and the respective security mechanisms of two well-known public cloud storage services—Amazon S3 and Google Cloud Storage—to demonstrate the fundamental applicability of our models (section 5). As we will see, our approach has the potential for making considerations about cloud storage security more structured, thorough, and comparable. First of all, however, we present some background motivating our deliberations and related work.

## 2 BACKGROUND AND RELATED WORK

In this section we introduce background information needed for the understanding of this paper. Therefore, we start by describing public cloud storage (section 2.1). Afterwards, we briefly introduce the relevant legal background (section 2.2). Next, we describe a notion of economic agency theory (section 2.3). We close this section with related work (section 2.4).

## 2.1 Public Cloud Storage

Public cloud storage provides virtually unlimited capacity to users on demand over broad network access while it is, usually, paid per use. The presumably most prominent examples of cloud storage are Dropbox and Amazon S3. However, Dropbox and Amazon S3 are of fundamentally different nature. Dropbox—like Wuala, Microsoft OneDrive, etc.—primarily approaches end users and provides file synchronization as well as sharing functionality for direct file access. Services like Amazon S3, in contrast, are rather aimed at developers using them through an application programming interface (API) to implement own functionalities upon these services. In the following, we concentrate on this Infrastructure-as-a-Service (IaaS) publicly available cloud storage for which we herein use the term "cloud storage".

For cloud storage, we distinguish—similarly to NoSQL systems that are typically running in the background of cloud storage—at least four different types: cloud storage with a key-value, column-oriented, document, and relational data model. Additionally, there are some other data models like graph data models and so on which are, though, rarely available publicly as cloud storage. Amazon S3 and Google Cloud Storage are examples for cloud storage with a key-value data model. Prominent instances for column-oriented cloud storage are Google Cloud Datastore or Amazon DynamoDB. Amazon Relational Database Service and Microsoft Azure SQL-Database are cloud storages with a relational data model. Last, Microsoft Azure DocumentDB is an example with a document data model.

Cloud storage has typically two different Application Programming Interfaces (API): A "management" and an "access API".[1] Using the "management API", a user can administer the cloud storage, e.g., he can initialize the service and manage data, user accounts, and access restrictions. The "management API" is, therefore, usually accessible over a SOAP- and/or REST-based web service and often additionally via a web application. Such "management web applications" are, for example, the Amazon Web Services (AWS) Management Console or the Google Developers Console. The "access API", on the other side, is usually only realized as a web service or—in the case of relational cloud storage—via vendor specific database drivers. For instance, if a user starts a MySQL instance in AWS, he accesses the database via the MySQL database drivers with the cloud storage's private or public IP. Through the "access API" a user can use the cloud storage, i.e., manage the data, from an own application.

As cloud storage is accessible from nearly everywhere over the Internet and all data stored in it resides in the cloud storage provider's data center, it must be secured carefully. This results in different requirements for confidentiality, integrity, availability, etc. Therefore, the security challenges and threats for cloud storage security, generally speaking, stem from diverse security sub-disciplines like database, web application, web service, and general security engineering [2]. Hence, securing cloud storage for building secure applications on top of this is of outstanding importance.

## 2.2 Legal Requirements for Public Cloud Storage

From a legal perspective, security aspects of cloud storage especially arise with regard to data protection regulations. Data protection law is focused on the protection of the data subject's fundamental right for informational self-determination which can be infringed by collection, processing and use of personal data. In data protection law, particularly relevant roles are the data subject (the one who needs to be protected), the controller ("cloud user"), the processor ("cloud application provider"), and the subcontractor of the processor ("cloud storage provider").

Within cloud computing cases, the relation between a cloud user and a cloud provider is usually classified as a so called "processing on behalf of the controller" [3]. This processor-controller model assigns the fulfillment of the data subject's rights to the "controller". In particular, the data subject can assert to rectify, erase, and block data which has been collected concerning him or her from the controller. The legal concept of "processing on behalf of the controller" furthermore requires the controller to supervise his processor concerning the compliance with data protection regulations. This means that the controller not only needs to verify that required

[1] This distinction is not always selective, as the API is often indistinguishable. However, it is a conceptual distinction of the functionality of the API. The distinction is based on the Storage Networking Industry Association for the Cloud Data Management Interface in [1].

technical and organizational measures are actually taken by the processor before the data processing begins but also as long as the "processing on behalf of the controller" continues. The fulfillment of all these rights and especially the data subject's rights are, however, hardly feasible for the controller in the context of cloud computing, as he is not in the position to reliably assess the actual conduct of the cloud provider. Real control options are therefore difficult to accomplish and mostly restricted, as on-site controls and inspections basically require excessive efforts [4] and cannot guarantee a proper conduct of the cloud provider with the data of the data subject. This problem is all the more intensified in cases where cloud providers are using subcontractors, as it becomes increasingly invisible to the cloud user what happens with the data of the data subject. The use of subcontractors itself is only legally admissible within data protection law if the controller agrees in writing to the processor's use of subcontractors. Even then, however, it has to be ensured that the legal requirements of the concept of processing on behalf of the controller are also respected within the sub contractual relationship. The cloud user therefore also has to fulfill his control rights against the subcontractors of the processor. He still remains, also across such service levels, the one who is responsible for fulfilling the data subject's rights.

Therefore, matching cloud computing—and particularly cloud storage—to the legal model of processing on behalf of the controller is significantly challenging. A generic usage model of cloud storage like the one presented herein will clearly help clarifying the duties emanating from data protection law and assessing respective challenges related to security.

## 2.3 Economic Agency Theory and Public Cloud Storage

The inexpediency of established models for ensuring appropriate security in the context of cloud storage is by far not limited to the legal domain. Beyond the traditional security requirements, cloud-specific instruments also have to address a fundamental conflict of interests between the cloud storage provider and the cloud storage user: Basically, the main interest in maximizing—or at least achieving a minimum level of—security lies with the cloud storage user who might have to fulfill certain legal duties (section 2.2) or who for other reasons wants to ensure certain security criteria to be met. The cloud storage provider, in turn, accounts for the overall security architecture that stored data are subject to, holds the power to decide on the security-related efforts actually being made, and has—without further measures being taken—a clear incentive to minimize these security-related efforts. This, in turn, results in a fundamental conflict of interests between the cloud storage provider wanting to minimize his security-related efforts and the cloud storage user who would profit from higher efforts but who is not able to reliably verify the measures actually taken by the cloud storage provider. The same fundamental conflict of course also emanates in further relations, e.g. between a cloud storage user and his respective customers.

From an abstract perspective, cloud settings and the fundamental conflicts resulting from them can—not only in matters of security—be understood by means of economic agency theory [5], leading to three abstract challenges having to be solved [6]: Adverse selection, moral hazard, and hold-ups. Of these, a particular relevance for security can be identified for adverse selection, emanating from a prospective cloud storage user's inability to assess the security capabilities of the cloud storage provider and leading to a "race to the bottom", and for moral hazard, resulting from a cloud storage user's general inability to monitor the cloud storage provider's security-related conduct as well as relevant surrounding conditions, again leading to incentives for the provider to underinvest in security or even to exploit the cloud storage user's data. In addition to those aspects already arising in traditional security-related settings, any sustainable approach to cloud storage security must also appropriately address these challenges in order to make the use of cloud storage feasible especially for those cases where simply trusting the cloud storage provider in matters of security is no viable option.

## 2.4 Related Work

There are several studies on security of and threats for cloud storage and cloud computing in general: For example, there are studies by the Federal Office for Information Security [7], by the Fraunhofer Institute for Secure Information Technology [8], and various other studies on cloud computing security in general [9], [10], [11], [12], and [13]. However, all these studies do not focus on publicly available IaaS-based cloud storage. As we lay out in the next sections, such cloud storage has specific properties and potential to be modeled in more detail in a generic

usage model containing the perspectives of different disciplines (section 3). Based on such a usage model a more sound threat model for cloud storage can be derived (section 4). To our knowledge, this is the first paper/study building such an interdisciplinary usage model and based on this threat model of public IaaS cloud storage.

# 3 A GENERIC USAGE MODEL FOR PUBLIC CLOUD STORAGE

As laid out above, security, legal, and economic aspects of cloud storage are of outstanding importance for the engineering of secure and legally compliant cloud-based applications on a profitable basis for the user and the cloud storage provider. In order to allow for well-structured interdisciplinary deliberations on cloud storage, we propose a generic usage model for cloud storage which integrates the different perspectives—the security, legal, and economic perspective—on cloud storage. The generic usage model is depicted in Figure 1.

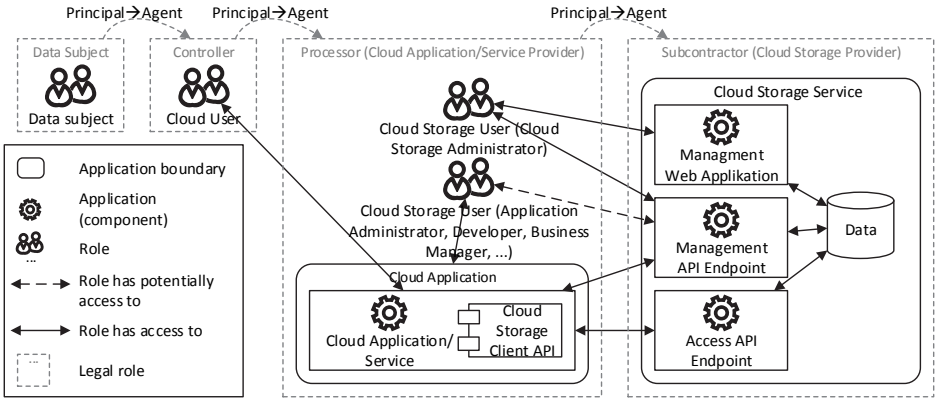


**Figure 1: Generic Usage Model for Public Cloud Storage**

Cloud storage systems running at a cloud storage provider's data center usually expose a "management" and an "access API" as well as a "management web application" (section 2.1). A cloud storage user employing the cloud storage of a specific provider, therefore, has to initialize the cloud storage or database over the "management web application". The initialization and initial administration of the cloud storage is done by a specific role. The "cloud storage administrator" usually has access to a root account—similar to the root account in operating systems. After initialization of the service, he can create subaccounts for other administrators, users, and applications. The further administration of the cloud storage can also be done via the "management API". For example, many functionalities of the AWS Management Console are also available directly via web services, i.e., the "management API endpoint".

To use the cloud storage within an application, cloud storage users have to utilize the "access API" of the provider. As mentioned in section 2.1, the "access API" is typically realized as a web service or it is a specific database driver. Many cloud storage providers offer downloadable and ready-to-use client API or command line tools. This "cloud storage client API" forwards the requests of the application to the web services—or the endpoints—of the cloud storage.

A cloud storage user implements an application or a service on top of the cloud storage ("cloud application"). For this, different roles can be provided, e.g., "administrators", "developers", and "business managers" which sometimes need own user accounts with specific access rights to the cloud storage. The application/service, then, is used by the "cloud user". In addition, the application may provide different further application roles. For instance, if the application is a human resource management application, there may be roles like "application manager", "case officer", "case handler", etc., which is, however, out of scope of our model. Within the implemented application, in turn, the "cloud user" may create and manage data about "data subjects" which are protected by data protection law. From the legal perspective laid out in section 2.2, the "cloud user" is thus the controller while the "cloud application/service provider" using the cloud storage is the processor and the "cloud storage provider" is a sub-contractor of the processor. Furthermore, every legal role is in a principal-agent-relationship with another legal role, as every role is in a contractual relationship with another. For example, the cloud storage user commissions the cloud storage provider to store his data in the cloud storage provider's data center (section 2.3).

# 4 A BASIC THREAT MODEL FOR PUBLIC CLOUD STORAGE

Having specified a generic usage model for cloud storage in the previous section, we now derive a basic threat model that can be extended for specific use cases and then can be used for threat analyses. For building the threat model, we used Microsoft's Security Development Lifecycle (SDL) that is described in more detail, e.g., in [14]. The resulting threat model is depicted in Figure 2.

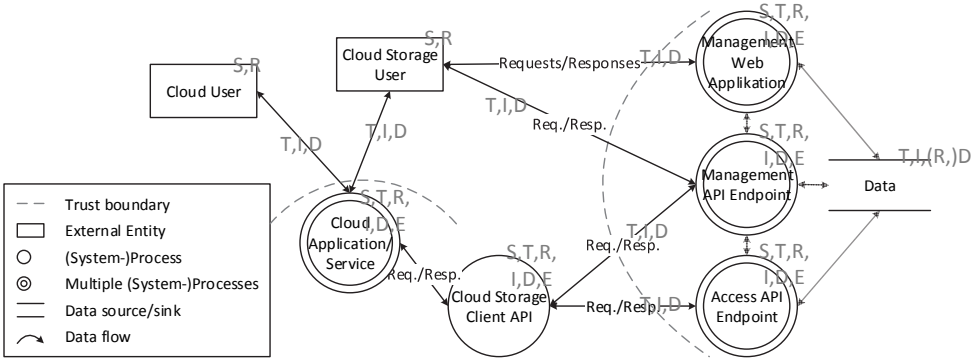


**Figure 2: A Threat Model for Public Cloud Storage**

A SDL-based threat model is based on a data flow model which consists of five basic elements: (System) Processes, external entities, data sources/sinks, trust boundaries, and data flow between all these entities (see: legend in Figure 2). Afterwards, every element of the data flow model can be analyzed for generic threats. Therefore, the SDL uses the STRIDE approach [14]. STRIDE is an acronym for the generic threats "spoofing", "tampering", "repudiation", "information disclosure", "denial of service", and "elevation of privileges".[2] These generic threats may occur in different model elements. For example, "spoofing" may occur in processes and external entities, i.e., in the processes "management web application", "management API endpoint", "access API endpoint", "cloud storage client API", and "cloud application/service" as well as in the external entities "cloud storage user" and "cloud user".

To apply this basic threat model to specific use cases—e.g., building a human resource SaaS-application based on Google Cloud SQL or a video streaming portal for classic music based on Amazon S3—, we then have to adopt the basic threat model. Furthermore, we have to consider the specific threats for this use case and the specific cloud storage. In doing so, we may extend the process "cloud application/service" as well as the external entities "cloud storage user" and "cloud user" with further processes and external entities. Additionally, we have to consider the security mechanisms of the cloud storage. Thereby, many generic threats can be mitigated by standard security mechanisms. For example, the threats "information disclosure" and "tampering" at the data flow—i.e., the communication link—between the "cloud storage client API" and the "access API endpoint" can be mitigated by SSL/TLS as a security mechanism.

# 5 THREAT ANALYSIS OF AN EXAMPLE USE CASE

In this section, we apply the basic threat model to an example use case, and carry out a threat analysis for the use case. For the implementation of the use case, we therefore imagine that the Java implementation is based on Amazon S3 (section 5.1) or Google Cloud Storage (section 5.2). In the following, we assume that we are building the, already mentioned, video streaming portal for classic music.[3] The streaming portal allows users to register and, afterward, to log in and listen to concerts. Therefore, the high definition videos and audio streams are stored at Amazon S3/Google Cloud Storage. A user has to pay a monthly fee or a fee for the access to selected content via credit card. As a consequence, we have to secure the user data and the access to the cloud storage since attackers may misuse credit card information of users.[4]

[2] For more information see, for example: http://www.microsoft.com/sdl/

[3] This example is inspired by the AWS use case Digital Concert Hall (https://www.digitalconcerthall.com) of the Berliner Philharmoniker. The story is described in more detail at: https://aws.amazon.com/de/solutions/case-studies/bph/.

[4] For reasons of simplicity, we assume that the user data is also stored in Amazon S3/Google Cloud Storage. In real use cases, highly structured user data more likely is stored within a column-oriented store or a relational database.

For reasons of space, we concentrate on the threat “information disclosure” at the process “access API endpoint” as well as the data flow between the “cloud storage client API” and the “access API endpoint” (Figure 2) for the threat analysis. Additionally, we assume that we trust the cloud service provider and thus, for the time being, exclude threats arising from the agency situation between the provider and the user laid out above. To better understand the points where the threat “information disclosure” may occur, we shortly describe an example use case and map out the typical data flow for the example use case “user registration and log in”. When a user registers at the streaming portal, a new user account is created at the client (“cloud storage client API”) containing, e.g., sensible credit card information. The data are then transferred to the cloud storage (“access API endpoint”) via a web service invocation (data flow between the two processes). Afterwards, the data are stored in the cloud storage and, then, can be accessed by the client and other clients connecting to the cloud storage to log in a user at the streaming portal.

Thus, considering concrete threats of information disclosure at the process “access API endpoint” we can derive the following four threats:

1. An attacker may passively eavesdrop the communication between the “cloud storage client API” and the “access API endpoint”. For instance, an external attacker may passively eavesdrop the data. Hence, data may be leaked to the attacker (Threat I1).
2. An attacker may actively eavesdrop the communication initiated by the “cloud storage client API”, e.g., by spoofing the “access API endpoint” (man-in-the-middle attack). As a consequence, the data may be leaked to the attacker (Threat I2).
3. An attacker may access the stored data as an authorized user (Threat I3).
4. An attacker may access the stored data as an unauthorized user (Threat I4).

## 5.1 Information Disclosure to External Attackers: Security Mechanisms of Amazon S3

In order to mitigate threats I1 and I2, the communication between the AWS Java Standard Development Kit (SDK) (“cloud storage client API) and the Amazon S3 web services (“access API endpoint”) is secured by SSL/TLS by default [15]. As the AWS web services are authenticated by their public key, a man-in-the-middle attack is alleviated. However, it is hard to implement public key/certificate pinning for some SDK to reduce the threat I2 even more.[5]

Additionally, every request sent to the AWS web services must be authenticated. Therefore, AWS web services use a specific way to sign every request (Signature Version 4). In Signature Version 4, the requests are signed with a secret access key id, a secret access key, and some other parameters like a timestamp [15]. This prevents replaying requests of authenticated users. Alternatively, it is possible to grant temporary access to Amazon S3 using another AWS service. For authorizing users to access data in Amazon S3, we can use bucket and user policies as well as access control lists (ACL) [15]. Bucket policies and ACL are resource-based authorization mechanisms. Therefore, policies are attached to Amazon S3 resources like buckets and objects. User policies, on the other side, can be attached to user accounts, groups, and roles to grant access. In combination, these mechanisms mitigate the threats I3 and I4.

To protect the stored data against unauthorized users (threat I4), AWS additionally proposes server-side encryption of the data. Using server-side encryption, data are sent to the web services and are then encrypted. There are three different types of server-side encryption: Server-side encryption with Amazon S3-managed keys, server-side encryption with AWS Key Management Service-managed keys, and server-side encryption with customer-provided keys [15]. For the first two types, AWS provides and manages the encryption keys. For the latter one, the customer uploads an own encryption key to AWS. However, the server-side encryption does not protect the data from an external attacker gaining access as an authorized user (threat I3), as the data are decrypted automatically on access for authorized users. Furthermore, server-side encryption is not able to secure the data against AWS employees or other attackers having access to the encryption keys. So, server-side encryption is not able to diminish any principal-agent-related problems. In addition to the server-side encryption, AWS provides an already

[5] Here, we refer to the AWS Forum: https://forums.aws.amazon.com/thread.jspa?threadID=157964

implemented client-side encryption in some SDK like the Java SDK. In client-side encryption, data are encrypted before sent to the servers of AWS which mitigates the threats I3 and I4 even more.

### 5.2 Information Disclosure to External Attackers: Security Mechanisms of Google Cloud Storage

For accessing Google Cloud Storage from a client ("cloud storage client API"), we can choose between different Java API: A JSON API and a XML API client [16]. Both communicate securely over SSL/TLS with the Google web services. This mitigates the threats I1 and I2.

For authentication in Google Cloud Storage, we can use three different mechanisms: OAuth 2.0, cookie-based, and service account authentication [16]. Each mechanism is recommended for different authentication use cases. The OAuth 2.0 authentication is recommended for authentication on behalf of a user, e.g., for tools and applications that are provided to other users. Cookie-based authentication is a browser-based authentication for, e.g., authenticated downloads. The service authentication is recommended for server-to-server applications like, e.g., a virtual machine running on the Google Compute Engine wants to access the data stored in Google Cloud Storage. To restrict access, Google Cloud Storage provides three mechanisms: ACL, Signed URL, and Signed Policy Documents [16]. Similar to Amazon S3, ACL in Google Cloud Storage clients allow the cloud storage user to grant access to other user accounts and groups. Using Signed URL, a user can grant time-limited read or write access to anyone in possession of the URL. Signed Policy Documents provide a way to specify what can be uploaded to a bucket. Thus, Signed Policy Documents, an enhancement of Signed URL, allow to specify parameters like size, content type, and other upload characteristics which are checked when visitors upload files to Google Cloud Storage. In combination, these mechanisms mitigate the threats I3 and I4.

In Google Cloud Storage, all data stored is encrypted by default using server-side encryption [16]. Like it is the case for Amazon S3, this does again not hinder attackers who gain access as an authenticated user (threat I3) and so on. Client-side encryption is, in contrast to the Amazon S3, not implemented in any Google Cloud Storage client API. However, a client API user can implement this feature easily on his own.

## 6 SUMMARY AND OUTLOOK

So far, we proposed a generic usage model for publicly available IaaS cloud storage with integrated legal and economic perspective and depicted a method for conducting a structured threat analysis based on this model. We believe that on this basis a more structured investigation of security, legal, and economic challenges in the context of cloud storage can be discussed. As the cursory threat analysis for the threat of "information disclosure" to external attackers in cloud application implementations based on Amazon S3 and Google Cloud Storage already show, our proposed model and method are practically applicable.

Even at this early stage of development, we thus foresee our proposed model and method to serve various goals for future research and development on security, legal, and economic aspects of cloud storage. First and foremost, our approach shall, like any modeling scheme, foster structuredness, completeness, and explicitness and provide a basis for communication among different stakeholders across different roles and disciplines. It shall thereby heighten the quality of security assessments as well as of respective engineering activities for settings significantly involving cloud storage. Especially our generic usage model will presumably prove valuable here as it encompasses those actors, relations and architectural entities specifically relevant in the context of cloud storage.

Besides constituting a structural basis for security analyses and respective implementation activities, our model and method may also help cloud users in demonstrating the thoroughness of their security-related considerations and precautions to external parties including their customers as well as regulatory authorities. Furthermore, having in mind the fact that cloud users are typically the controller and thus bear primary responsibility in matters of data protection law (section 2.2), our approach might also help them to better select appropriate cloud storage providers and to better identify their legal responsibilities.

This does, however, also lead us to the current opportunities for improving our model which we are going to address in the future: First of all, we have to prove and refine our models in much more use cases. Furthermore, the cloud storage provider itself is currently not explicitly addressed as a potential adversary within our usage model. As delineated in section 2.3, as recognized with regard to the value of server-side encryption in sections 5.1 and 5.2, and as also underlying many legal regulations—including, in particular, the controller's verification obligations mentioned in section 2.2—, however, security threats might also arise with regard to a storage provider being not (absolutely) trustworthy. For the future, we thus plan to explicitly integrate the storage provider as a potential adversary into our threat model, too. Last but not least, we also plan to undertake further research integrating the three disciplines—cloud storage security engineering, law, and economic agency theory—on, for example, new legally compliant security mechanisms and the security implications of nested principal-agent-relationships.

## REFERENCES


[1] Storage Networking Industry Association (2014). Cloud Data Management Interface (CDMI) – Version 1.1.0. http://www.snia.org/sites/default/files/CDMI_Spec_v1.1.pdf.

[2] Winkler, V. (2011). Securing the Cloud–Cloud Computer Security Techniques and Tactics. Waltham.

[3] Art. 29 Data Protection Working Party (2010). WP 196, Opinion 05/2012 on Cloud Computing.

[4] Heidrich, J.; Wegener, C. (2010): Sichere Datenwolken - Cloud Computing und Datenschutz, MMR pp.803-808.

[5] Hauff, S.; Huntgeburth, J.; Veit, D. (2014). Exploring uncertainties in a marketplace for cloud computing: a revelatory case study. Journal of Business Economics, 84(3), pp. 441-468.

[6] Pallas, F. (2014). An Agency Perspective to Cloud Computing. Proc. of the 11th Intern. Conf. on Economics of Grids, Clouds Systems and Services (GECON), pp. 36-51.

[7] Bundesamt für Sicherheit in der Informationstechnik (2012). Überblickspapier Online-Speicher. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Grundschutz/Download/Ueberblickspapier_Online-Speicher_pdf.pdf?__blob=publicationFile.

[8] Fraunhofer Institute for Secure Information Technology (2012). On the Security of Cloud Storage Services. https://www.sit.fraunhofer.de/fileadmin/dokumente/studien_und_technical_reports/Cloud-Storage-Security_a4.pdf.

[9] Cloud Security Alliance (2011). Security Guidance for Critical Areas of Focus in Cloud Computing V3.0. https://cloudsecurityalliance.org/guidance/csaguide.v3.0.pdf.

[10] Cloud Security Alliance (2013). The Notorious Nine - Cloud Computing Top Threats in 2013. https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf.

[11] European Network and Information Security Agency (2015). Cloud Security Guide for SMEs. https://www.enisa.europa.eu/activities/Resilience-and-CIIP/cloud-computing/security-for-smes/cloud-security-guide-for-smes.

[12] Open Security Architecture (2015). SP-011: Cloud Computing Pattern. http://www.opensecurityarchitecture.org/cms/library/patternlandscape/251-pattern-cloud-computing.

[13] National Institute of Standards and Technology (2011). NIST Guidelines on Security and Privacy in Public Cloud Computing. http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909494.

[14] Shostack, A. (2014). Threat Modeling: Designing for Security. John Wiley & Sons.

[15] Amazon Web Services (2015). Amazon S3 Developer Guide. http://docs.aws.amazon.com/AmazonS3/latest/dev.

[16] Google (2015). Google Cloud Storage – Documentation. http://cloud.google.com/storage/docs/overview.