This is a pre-print of the paper accepted to be published as part of
10th IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2018),
held in Nicosia, Cyprus on 10-13 December 2018.

C^{3}S: Cryptographically Combine Cloud Storage for Cost-Efficient Availability and Confidentiality

Leon Sell
Chair for IT-Security, University of Passau, Passau, Germany
sellleon@fim.uni-passau.de

Henrich C. Pöhls
Institute of IT-Security and Security Law (ISL)
Chair for IT-Security, University of Passau, Passau, Germany
hp@sec.uni-passau.de

Thomas Lorünser
AIT Austrian Institute of Technology, Vienna, Austria
thomas.loruenser@ait.ac.at

October 15, 2018

Abstract

Increasing the availability by using multiple cloud storage providers for replication comes at a price; not only does it increase storage costs with every replica, it also greatly disperses the information to different cloud storage systems. Thus, all storage locations must be trusted to **not** read that data. Contrary the cryptographic technique of secret sharing splits data into confidentiality protected *shares* and if the adversary does not have access to more than a pre-defined threshold k of those shares, then the data's confidentiality is protected. At the same time secret sharing also increases the availability because the legitimate user must only download the data from k out of n shares. The goal of this paper is to quantify the economic advantages of efficient and secure information dispersal strategies in multi-cloud settings based on the current market situation. Therefore, we put together a database of 63 cloud storage offers and analyzed opportunities to combine them into virtual storage services delivering availabilities of 99.999% at the best price. Additionally, the combined multi-cloud storage is leaning towards data protection legislation of the European Union (EU), as any combination of k shares includes at least one from an EU-based provider. This inhibits non-EU jurisdictions to 'subpoena' the required number of shares to reconstruct data without the help of an EU-based provider. Our findings show that it is possible to find combinations which give the cloud storage consumer the wanted high availability and legal compliance guarantee at half the cost of any two providers from within the EU storing unencrypted replicas.

1 Introduction

Cloud storage has become a commodity technique, commonly used by companies to dynamically outsource their data storage onto third-party servers. Benefits include increased agility leading to decreased monetary costs, access to managed storage without having to employ storage specialists as well as improved off-site disaster recovery. Indeed we had no problems finding 63 different offers for storage as a service. The differences of the offers we recorded are their geographic location, their availability and last, but not least, their price. For all of them we used the advertised values, for which the cloud service consumer would be given contractual agreements, i.e. service-level agreements. Our goal was to get a high availability, e.g. above 99.999%, for a reasonable price. However, with the outsourcing come some drawbacks: increased dependency upon third-parties, vendor lock-in, loss of data sovereignty and privacy. Especially the latter, are confidentiality problems that our solution wants to overcome. While companies would like to reap the monetary benefits, data confidentiality issues prevent cloud adaption, esp. with the new, potentially existence-threatening, fines contained in the upcoming General Data Protection Regulation (GDPR) of the European Union [1].

A new approach that allows to mitigate some of those problems is the *cloud-of-cloud [2]* or *multi-cloud* paradigm. Here, the storage system consists of multiple independent cloud storage providers; this technique disperses the consumer's data redundantly over multiple independent storage clouds, thus limiting the damage potential of an outage of each single storage provider. However, the privacy problem remains—the cloud storage provider is still *honest-but-curious*[1].

To overcome not only the availability but also the confidentiality problem we propose to combine the multi-cloud storage with secret sharing—a technique to achieve cryptographically proven confidentiality protection, i.e. prohibit each single provider of learning the stored contents. The cryptographic technique of secret sharing—invented by adi Shamir in 1979 [3]—splits data into n so-called *shares* and if one does not have access to a pre-defined threshold k of those shares the data's confidentiality is protected. This cryptographically guarantees the confidentiality against a single storage provider which only holds 1 of n shares—and even against $k-1$ colluding providers.

Together this gives confidentiality for highly-available storage at a very competitive cost, if more efficient protocols are used than the original Shamir method. Fig. 1 shows the significant cost savings, e.g. more than 50% for an availability of 5-nine[2], we got when comparing 63 different offers using this combined approach compared to plain replication. Also none of the providers in our market analysis offered such a high availability as a direct commodity product.

Note, this combined multi-cloud storage solution is cryptographically enhanced and thus provides the consumer with technical rather than otherwise only contractual guarantees: Using secret sharing increases data confidentiality under a non-collusion assumption between the involved storage providers. Secret sharing cryptographically guarantees the confidentiality against an attacker that has less than k of n shares. At the same time it increases the availability because the legitimate user must only download the data from k out of the

[1] A cryptographic term describing an adversary that collects all information it learns during interactions but behaves according to protocol.

[2] This is a shorthand often used for above 99.999%.

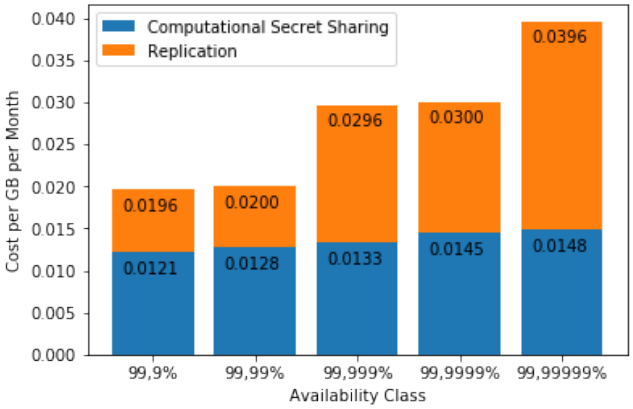

Figure 1: Cost per GB using Secret Sharing vs. Replication (optimal k & n for $k \leq 10$)

overall n shares, i.e. $n - k$ storage locations can be unavailable and the data can be reconstructed from the remaining k shares.

Finally, it saves storage space as each share is of factor $1/k$ of the data that shall be stored. In summary, if confidentiality and availability are contradicting goals in the design of the multi-cloud storage system based on secret sharing the selection of good provider configurations and parameters is more involved compared to standard replication based approaches.

The contribution and outline of the paper is as follows:

- In Sec. 2 we discuss the current approaches and solutions for increased availability and confidentiality as well as the state of the art.
- In Sec. 3 we provide details on what we think is the first comprehensive cloud storage database comprising pricing, availability and geolocation information. The database is open access[3] and serves as a first basis for cloud brokerage algorithms that are interested in these metrics. We discuss why these three properties, which can be found in legally-binding service level agreements (SLAs), are interesting for a cloud consumer interested in availability as well as data protection[4] aspects.
- In Sec. 4 we provide some details on a capability modelling approach, which models key service level agreement parameters, such as cost, geolocation and availability formally—using Haskell—and allows to calculate them for multi-cloud storage solutions that employ secret sharing, which we also formally modelled.
- In Sec. 5 we present our decision support system which helps in the selection of cloud storage, with a special emphasis on secure dispersed multi-cloud configuration.
- In Sec. 6 we discuss the results and conclude in Sec. 7.

[3] https://github.com/Archistar/c3sp/blob/master/StorageProviders.json

[4] Note, the term data protection is also known as privacy or as protection of personally identifiable information (PII).

2 Challenges and Approaches for High Availability

Two main concepts have been used over the last decades to protect data in storage systems from loss and to achieve high availability. On the lowest layer RAID (Redundant Array of Independent Disks) technology has been used to protect from hard disk outages or bitrot and replication has been used to prevent from large scale failures.

Given the huge amount of data acquired and stored today and the architecture of modern data centers, this combined approach does not fit well anymore, nor does it provide the required cost effectiveness. Especially the flexibility of RAID technology is very limited, because it works on block level and for large drive sizes of multiple Terabyte rebuilding after failures could take days. In combination with current drive failure rates this leads to unacceptable situations where many additional disk are needed. Additionally, replication introduces a lot of storage overhead if more than one replica is needed compared to the approaches discussed below. Furthermore, the current trend towards outsourcing, e.g., like cloud computing, also impacts the design of modern storage systems and protocols, because cloud users do not have access to the underlying hardware anymore and higher layer (overlay) protocols are needed to reduce provider dependency and lock-in.

2.1 Availability and Erasure Coding

In general, we see a major trend away from dedicated monolithic storage solutions of specific vendors towards the use of cheap commercial off-the-shelf (COTS) hardware and/or services to reduce cost and prevent from vendor lock-in. However, because of the high failure rate of COTS hardware, an additional layer of redundancy on top is needed to get the desired reliability and availability of the overall storage system.

The application of *erasure coding* is considered the most efficient solution to get high availability in modern storage solutions [4]. The basic idea of erasure coding is to encode data into multiple blocks which are then stored on different disks. However, because only a predefined subset of blocks is needed for reconstruction the overall availability and durability of the system is increased. Compared to replication erasure coding can be much more storage efficient for larger storage systems and is currently making its way into all modern large scale storage systems[5,6,7].

Although the approach is very interesting and achieves good efficiency, it does not deal with data privacy, which is another important aspect, specifically if the configuration involves third party storage providers. The only way to consider confidentiality in this scenario is to add an additional layer for encryption on top of the storage layer. Although this approach is straight forward, it introduces additional difficulties in the management of the keys and could also negatively impact the performance. Additionally, if the key is lost, the data cannot be recovered anymore and the overall availability crucially depends on

[5] https://www.gluster.org, accessed July 2018.
[6] https://ceph.com, accessed July 2018.
[7] https://docs.openstack.org/swift/latest/, accessed July 2018.

the availability of the key. Therefore, it would be desirable to gain availability and confidentiality for data already in the encoding layer for best efficiency and flexibility without additional key management overhead.

2.2 Security for Dispersed Data

An alternative approach to erasure coding is *secret sharing*. It builds upon the concept of confidentiality-protected data dispersal in a distributed system. In the following we quickly introduce the most important encoding schemes in storage.

For storage it is important that encode/decode steps are computationally efficient and the size of the fragments is optimally small for the required security properties. Furthermore, secret sharing schemes with threshold access structures are the most efficient and practical option for storage systems. Threshold secret sharing implies that data is encoded and split up into n fragments called shares/chunks that independently do not reveal any information about the original content and any arbitrary subset of k chunks ($k \leq n$) can be used to recover the data. k is called the threshold and can be freely chosen at encoding time within the given range. Thus, if k is smaller than n and a certain data chunk is lost or not accessible, the data owner is still able to regain the information by gathering other chunks which gives the increased availability. In that sense, secret sharing is very similar to erasure coding, which also provides the k-out-of-n decoding properties, however, contrary to erasure coding it also provides security guarantees in form of confidentiality. Based on secret sharing a secure storage can be built, if the chunks are subsequently distributed to separate cloud service providers (CSP) which do not collude, i.e., they do not share the chunks they are holding. Thus, the confidentiality of data is maintained as long as no more than $k-1$ clouds collude, and the data will remain available so long as k out of n chunks are accessible at the same time.

Perfectly secure secret sharing (PSS) was introduced by Shamir [3]. Shamir's scheme is well suited for both, software and hardware implementation, especially if small message spaces can be used and the chunk size is optimal. It is perfectly secure, because it gives information theoretical security guarantees, i.e., when the receiver has less than k shares available even computationally unbounded adversaries are not able to recover the plaintext. However, although the share size is optimal for perfect security, still each share needs to be as long as the original file, which makes the scheme very expensive for storage applications. In fact, the scheme is even less storage efficient than pure replication.

To enable more efficient storage solutions Information Dispersal (IDS) as introduced by Rabin [5] is used, which is equivalent to non-systematic erasure codes [6]. IDS basically apply a non-systematic erasure code to generate data chunks, and therefore does not give any security guarantees. Nevertheless, it produces the shortest possible fragments together with a k-out-of-n access structure and the integrity properties are identical to the PSS scheme. Therefore, if both schemes are combined, Computational Secret Sharing (CSS) can be build. The combination was first proposed by Krawczyk [7] and enables efficient and secure storage of data with computational security and chunks by a factor of $1/k$ shorter compared to PSS. Although CSS is not perfectly secure it still provides strong security guarantees and can be also considered quantum-safe.

Besides the plain secret sharing modes which only resist less than 1/3 of

erroneous shares in the reconstruction step robust versions have also been proposed which work as long as the majority of the shares in the reconstruction process are correct. The robust PSS version is based on information checking techniques by Bishop et al. [8] and for CSS it uses the simpler fingerprinting as proposed by Krawczyk [9].

2.3 Related work

RACS [10] was the first system leveraging erasure-coding to distribute data over multiple storage clouds. Their main goal is to prevent vendor lock-in and to achieve high availability; privacy concerns are not discussed. It mimics the Amazon S3 interface for communication with its clients and if a single RACS installation becomes a performance bottleneck, distributed RACS can be deployed. RACS was also the first to analyze the economical benefits for dispersed cloud storage and also did a first analysis of the economical benefits. However, it only focuses on economic failures and cost, but does not consider security constraints for confidentiality and integrity nor treat availability in a comprehensive way.

HAIL [11] is another approach which focuses upon high-availability and integrity protection within the cloud; also here data privacy is not of primary concern. To achieve high availability, data is distributed (using erasure codes) upon multiple clouds and data on a single server has additional redundancy attached to increase resistance against bitrot.

A more recent approach also considering data confidentiality is DepSky [12]. It offers an object-store interface on top of passive storage clouds. Its data objects utilize cryptographic hashes for integrity control and short-time version numbers provide for concurrent updates. It also has limited support for concurrent writers through client-side locks. Confidentiality is optionally supported by secret-sharing techniques in the DepSky-CA variant.

A very flexible framework for secure multi-cloud has been presented in [13]. It is called ARCHISTAR and provides different option and technologies to build secure distributed storage systems. It supports various kinds of secret sharing techniques as well as an optional Byzantine resilience layer. Core algorithms in Java and JavaScript are available as open source[8] and have been the starting point for our analysis.

In general, we see more and more approaches leveraging the multi-cloud paradigm to protect data from loss, increase availability as well as for confidentiality and integrity reasons. It is secret sharing solutions, which provide all the desired features without introducing complex key management and although we expected them to make its way into commercial solutions their application is still hampered by the complexity in the configuration step. For systems to be successfully deployed, they need to be trusted and perceived as valuable and usable by different stakeholders such as the end-users, managers, and SLA responsible officers [14,15] and this is exactly where analyzed solutions fall short. The technical foundations are well understood, but do not solve the problems encountered by operators in the deployment phase, which is an essential part in the life cycle of an application [16]. The solutions provide too many degrees of freedom in configuration. From our initial analysis we saw, that the adoption of secure multi-cloud storage is hampered by missing guidance on the user

[8] https://github.com/archistar, accessed Oct. 2018.

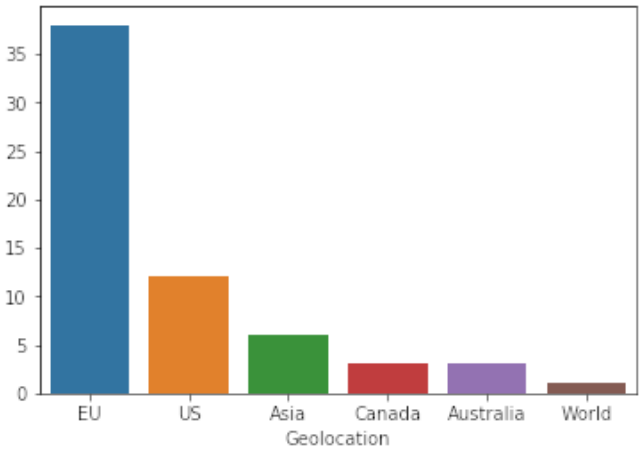

Figure 2: Geolocation distribution of chosen cloud storage providers

side and a lack of understanding of their needs when it comes to security and data protection, especially in systems based on secret sharing which build their security on the non-collusion assumption.

To the best of our knowledge no work on this particular topic exists and we try to fill the gap by designing a model-based decision support service which is easy to use and relies on a comprehensive provider database with actual pricing information. It should support the user in cloud adoption for multi-cloud storage application by letting them design a virtual storage service as a multi-cloud storage network. For our work we assume the use of CSS algorithms, which are in most situations the natural choice for storage applications.

3 Database of Cloud Offerings

We have collected the key parameters for costs, availability and geographic location for 63 different cloud service offers. The database has been filled with business offers for commodity cloud storage for the following reasons: (1) In the commodity world we have open APIs and in the consumer world we have proprietary protocols and client software. Thus, the former allow to combine several cloud storage providers into a combined provider by using the API to effectively communicate with all the individual providers, e.g. using Amazon's S3 protocol, that form the combined storage solution. (2) In the consumer world the pricing schemes are based on bundling [17] which make comparisons very hard. (3) Many consumer offers are also often combined with other services not related to storage which further complicates the calculation of pure storage cost. On the contrary, business offers are based on block pricing models which support a pay-as-you-go idea of cloud computing and allow for a more accurate cost estimation and comparison.

3.1 Database entries

The database[9] contains 63 entries, see Fig. 3 for an example. For each offer we stored the following:

id Incremental identifier of the entry.

name Name of the cloud storage offer.

company Company that offers the cloud storage.

serverLocation Geolocation of the server, see Fig. 2.

encryptedStorage Encryption used by the company to secure stored data.

availability Availability of the cloud storage.

cost Direct and related costs, e.g. *costPerGBStorage* but also *costPerGBInboundTraffic*, or *costPerGetOperation*.

delayedFirstByte Binary value indicating whether data can be retrieved immediately from the cloud storage. Archive-tier cloud storage often has an access time from a couple of minutes to a couple of hours.

minStorageDuration Minimum storage duration for data stored on the cloud storage in days.

note Note containing additional information.

```
id:                         0
name:                       "Google - Multiregion EU"
company:                    "Google"
serverLocation:             "EU"
encryptedStorage:           "AES-256"
availability:               0.9995
accessLocation:             "EU"
cost:
  costPerGBStorage:         0.026
  costPerGBInboundTraffic:  0
  costPerGBOutboundTraffic: 0.12
  costPerGetOperation:      4e-7
  costPerInsertOperation:   0.000005
  costPerDeleteOperation:   0
  costPerListOperation:     0.000005
  costPerUser:              0
delayedFirstByte:           false
minStorageDuration:         0
note:                       "Always Free:\nRegional Storage\t5 GB-months\nClass A
                            Operations\t5,000\nClass B Operations\t50,000\nNetwork
                            Egress\t1 GB from North America to each GCP egress destination
                            (Australia and China excluded)"
```

Figure 3: Example database entry

[9] https://github.com/Archistar/c3sp/blob/master/StorageProviders.json

3.2 Cloud storage market with standardised offers

We would like to acknowledge that it can be hard to compare cloud service levels and therefor the different cloud storage offers on the market. As an example take the definition of availability from ISO 19086-1 [18], which is an international standard on service level agreements for Cloud Services: "The availability component specifies the method for determining that the covered services are accessible and usable." [18] As a result one gets a quantitative, thus comparable number. However, ISO 19086-1 also acknowledges that it is not easily comparable stating: "There may be cases, such as 'scheduled downtime' where the cloud service is not available for reasons other than failures." [18] which might not negatively impact the availability. Hence, details matter and might hinder an easy comparison.

However, availability is of interest to cloud service consumers and thus a standardised component of a service level agreement, e.g. internationally standardised in ISO 19086. In ISO 19086-4 [19][10] one finds more security as well as data protection relevant properties of cloud services. For this paper we have concentrated on supporting the cloud storage consumer to select a set of storage providers to optimise five aspects:

- *confidentiality* against the storage provider[11] as a prerequisite for increased security,
- *availability*,
- *costs* as the economic aspect of storage, and
- *geographic location of data*[12] as a data protection aspect.

We will explain them and their modelling in Sec. 4.

3.3 Generation of an SLA for a combined service

The solution for increased availability and confidentiality is a combined service, hence it is a multi-cloud or federated cloud service. With standard APIs the n shares are in need to be distributed to n different cloud storage providers. An example of an entry in our storage provider database is Google's offer termed "Google - Multiregion US" for which we note that the company is '`Google`' and the 'serverLocation' is '`US`'. It offers an availability of `99.95`% at a cost of `0.026` USD/GB. So for each share the user configuring the secret sharing service would need to find a suitable storage provider. However, as storage is offered as a commodity with a standard API it could be any of the entries in the database, e.g. "Amazon - IA London" by '`Amazon`' located in '`UK`'. The latter offers an availability of `99.0`% at a cost of `0.01048` USD/GB. While choosing between the two of them could be simple assume those are combined in a simply replicated multi-storage without additional confidentiality into a multi-cloud service. Then the combined costs increase to `0.03648` USD/GB, while the availability increases to

[10] Note, ISO 19086 part 4 is not yet a final international standard, but at the time of writing it is in the final stages of the ISO standardisation process.

[11] Under ISO 19086-4 this would be listed as among the "Cryptographic controls for data at rest" [19].

[12] Under ISO 19086-1 this would be specified in the "Data location component" [19] of an SLA.

99.9995%. The problem is that now one has replicated the data to a location in the US and thus might need additional contractual agreements due to European data protection legislation.

Our idea is to model how different numbers for the amount of shares (n) and the reconstruction threshold (k) impact the actual availability, costs, and geolocation if one takes into account the different offers on the market.

Moreover, we enable SLA tailoring, i.e. the consumer can choose the intended results, for the multi-cloud configurations. This is very attractive, because the standard cloud storage market provides only limited flexibility in the configuration of service level agreements (SLA). In fact, serving standardized SLAs to customers is a major feature of cloud computing which helps to enable the elasticity and self-service capabilities the customers want to have.[13] In combination with cryptographic methods that combine several offers this quickly becomes a combinatorial nightmare, hence it is very difficult for customers to find offers which perfectly fit their particular needs and almost impossible to negotiate special conditions. In particular, the k-out-of-n paradigm is used to design systems which can theoretically provide arbitrary high levels of availability. Availability classes are typically given as number of leading nines of the availability value, i.e., a '3-nines' availability means 99.9% which corresponds to a downtime of $8.76h$ per year or $43.8min$ per month. We have used this type of availability classes to demonstrate not only a theoretical value, which one can reach in a system of secret shared storage, but a value that one could also buy on today's market of commodity storage in the cloud.

4 Model of costs, availability and geolocation of the virtual storage network

In order to design and implement a decision support system to help choose cloud storage providers for storing CSS shares, we model key properties in respect to the values of n and k for the threshold secret sharing. As discussed earlier, it is not trivial to find an optimal combination of cloud storages for one's needs as even a small amount of cloud storages to choose from will result in a large number of possible combinations. The cloud service consumer will supply information about his/her requirements—e.g. how much they want to pay or how much availability they require—which the decision support system will then use to find combinations of single cloud storage providers fitting the given criteria. We modelled the properties of availability, cost, and geolocation using Haskell, which was also used for implementing the decision support system.

Based on Haskell models for each property we built a model of the secret sharing cryptographic primitive. Recall that the secret is divided into n unique shares where the possession of any k or more shares enables the reconstruction of the secret.

4.1 Modeling Availability for Secret Sharing

The availability of a combined cloud storage provider refers to the percentage of time the user can reconstruct the original file from the shares stored on the

[13] Still in many cases main criteria like availability are not clearly stated in provider SLA, e.g. see Amazon S3 service https://aws.amazon.com/s3/sla/ accessed July 2018.

servers of the combined cloud storage provider. It is calculated from the single availabilities of shares. For this purpose the availability is interpreted as the probability of the server being reachable at any given time. E.g. an availability of 99.9% is equivalent to: there is a 99.9% change that the server is reachable at any given time. In order to calculate the comined availability one has to calculate the probability of having at least k reachable server at any given time.

The Haskell implementation calculates this by generating the binary probability tree of the servers being reachable or not and then finding all valid paths in the tree. A valid path is a path where at least k of the servers are available. The haskell implementation is optimized to the effect that the paths are only constructed to the point where the condition is either fulfilled or it is no longer possible to fulfil it. The probability of each valid path is then being calculated by forming the product of the probabilities of the edges of the path. Lastly the probabilities of all valids paths are summed which yields the overall probabilities.

4.2 Modeling Geolocation as a Data Protection Aspect

Consider the data to be stored being medical data about European Citizens. Then it might be a legal requirement that this data shall only be processed on cloud nodes which are actually physically located within the EU. In our model this translates to having no more than $k - 1$ shares on cloud storage servers which are physically located outside of the EU. This could be seen as the joint geolocation of a service that stores[14] data. In our hierarchical model of the geographic location the country labels of EU member states are below a level tagged as `EU`. See Fig. 2 for an overview of the distribution of the servers' locations in our database. Within the model we have encoded that for example a geolocation level of `DE` for Germany is compatible with that of the `EU`, and thus captured in the model that Germany is a EU member state. The labels can also be used to model data protection legislation for each storage provider, e.g. EU-US Privacy Shield or US Freedom Act.[15] Therefore, the calculated geolocation of a combined cloud storage provider does not refer to the geographical location of the servers. Rather the combined geolocation refers to the boundary in which the original file can be recreated. Meaning without a member from inside of that boundary there are not enough shares to recreate the original file. The geolocation property is predefined by the user, i.e. the query posed by the user.

4.3 Costs Model for Secret Sharing

We have modeled several different kinds of costs that a cloud storage provider could charge for. The different kinds of costs of the multi-cloud provider arise in different ways from the single servers. E.g. the storage cost is the sum of the single storage costs divided by k as each of the n shares is only of factor $1/k$ of the original data for CSS encoding. Inbound traffic is calculated in the

[14] Storing falls under 'processing' in EU data protection regulation.

[15] Note, we are aware that this can only be decided on a company level and not solely on a country level, as the Privacy Shield only refers to privacy principles that companies can voluntarily comply with. However, to fall under the EU-US Privacy Shield the storage must be located in the US. And being located in the EU lifts the burden of requiring the privacy shield.

same way. Outbound traffic, however, is the average outbound traffic cost of the single servers, as one only needs to retrieve k of the shares and the size of each share is only of the factor $1/k$ of the original data. Therefore the the formula is $\frac{s \times k}{n \times k}$ or $\frac{s}{n}$ where s is the sum of the outbound traffic costs of the single servers; in other words its the average outbound traffic cost.

5 Decision Support System

The user that wants to cryptographically protect their data from prying eyes of potentially colluding (or legally forced to collude) storage providers outside a certain geographic boundary and who wants to reach a certain availability can query our decision support system C^{3}S. The user defines the intended set of properties that the combined cloud storage providers shall have and thus defines the target function of the decision support system. The user can determine the properties described below. All of them are optional, a sensible default behaviour will be chosen if an option is not specified.

k — The amount of shares, i.e. providers, needed to restore the secret. If not given, the lowest possible value for each possible combination will be chosen. "lK" and "uK" can be used to define lower and upper bounds for "k" instead.

n — The total amount of shares, i.e. providers. If not given, any valid combination will be calculated. "lN" and "uN" can be used to define lower and upper bounds for "N" instead.

loc — The geographical boundary in which the secret is permitted to be restored. If not given, the boundary will default to `World`, meaning it is not restricted. Examples for location are: `BE`, `DE`, `EU`, `JP`, `US`, `World` and `Local`.

avail — The minimal availability any combination of providers is required to have. If not given, the availability is not restricted. E.g. `0.9999` for 99.99%.

cost — The maximum amount for each type of cost[16] the combined storage is allowed to cost in US Dollars (USD). If any of the subcosts are exceeded, the combination will be discarded. If not given, the amount of cost or subcost is not restricted.

limit — Limit can be used to restrict the amount of results that be returned by the target function. If nothing is specified, all possible combinations will be returned

delay — Specifies whether or not offers with first byte delays should be used. If not specified, all offers will be used.

minDur — The maximal minimal storage duration on offering is allowed to have. If not specified, all offers will be used.

order — The order in which the results will be returned. Can either be `"ByPerGBStorage"` or `"ByAvail"`. Default is `"ByPerGBStorage"`.

[16]The currently supported subcosts are: costPerGBStorage, costPerGBInboundTraffic, costPerGBOutboundTraffic, costPerGETRequest, costPerPUTRequest, costPerPOSTRequest, costPerLISTRequest.

5.0.1 Algorithm

The target function is responsible for carrying out the selection. It as a depth-first search algorithm that traverses a virtual binary decision tree were each decision is either using or not using a specific provider. Virtual meaning that the tree is not actually being constructed in memory. This means the algorithm has a theoretical worst case complexity of $\frac{N!}{n!(N-n)!}$ where N is the number of cloud storage providers in the database. However the typical run will have a much bettery complexity as most of the possible combinations will not be generated due to already having found cheaper combinations and already having $k-1$ shares outside of the specified geographical boundary.

The target function takes a list of single cloud storage providers and a query as input and produces a list of combined cloud storage providers. In Haskell, a list being a countable number of ordered values that allows duplicates.

The target function is a recursive function that traverses a list of jobs. A job contains a list of previously chosen cloud storage providers and a list of cloud storage providers that can be added to the first list. Initially the job list is a list with one job that has no previously chosen providers and the entire list of cloud storage providers from the database to chose from. Additionally the function keeps a list of the best multi-cloud configurations.

Each call will remove the first job from the list of jobs and process it. Up to two new jobs will be added to the job list each call: one where the first server from the lists of servers to choose from will be removed and added to the list of previously chosen servers, and one where the first server will be droped without adding it to that list. The first job will only be added if the first server from the list of servers to choose from together with the previously chosen servers can result in a valid multi-cloud configuration. Meaning if the new server together with the previously chosen servers would exceed the cost limit or allow the secret to be reconstructed outside the specified boundary, the job will not be added to the list of jobs. Additionally, if they already form a valid multi-cloud configuration, this configuration will be inserted into the list of best multi-cloud configurations and replace the previously worst entry in the list if the limit specified by the query is reached. The recursion will end when each job has no more elements in the list of servers to choose from.

6 Evaluation

As mentioned previously and as shown in Fig. 1, we achieve a cost reduction with a simultaneous increase in availability compared to using simple replication. When compared to other dispersed storage systems which also offer n-out-of-k the use of secret sharing allows to gain confidentiality against curious storage servers under a non-collusion assumption. As several papers on secret sharing have discussed, the overhead is considered practical [13, 20, 21], [22], both computationally and storage wise. With the decision support system C^3S presented in this paper we are able to find interesting configurations for the cryptographic mechanism's parameters in a realistic market.

For example we checked for an increasing threshold what the cost-effective (`costPerGBStorage`) combination of servers would be to reach above 99.999% (5-nine) availability. Note, for our results we always required no reconstructability

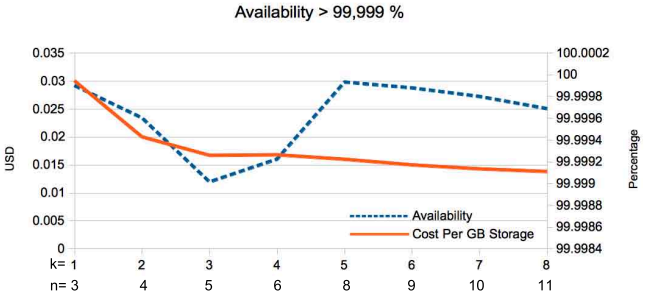

Figure 4: Availability in %; Storage costs in USD; k out of n; combined geolocation of EU

outside the EU, no first byte delay and a maximum of 30 days of minimum storage duration. The result is depicted in Fig. 4 and shows that we have a sweet spot at 5-out-of-8 for this particular example. However, due to real market's actual costs and the contractually offered availability going for higher n than the selected 5-out-of-8 selection is not suitable, especially as this combination gives even 99.9999% (6-nine) availability.

Note, as already visualised in Fig. 1 one can also see in Fig. 4 that three replicas (1-out-of-3) achieve 6-nine but are very costly as it contains only offers which have their servers located in the EU. For even better visualisation we have generated a heatmap of the costs per GB in the different availability classes as Fig. 5.

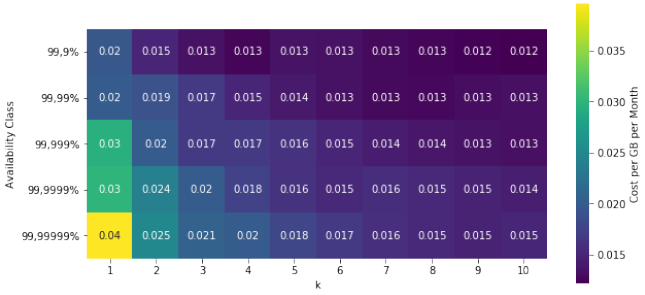

Figure 5: Cost per GB trend for different values of k and different availability classes (optimal n)

7 Conclusions

With the help of formally modelled properties of cloud storage providers in Haskell and a database of real market data we were able to provide a decision support system for selecting the most cost-effective storage providers from the market to provide a cryptographically confidentiality protected multi-cloud storage based on secret sharing. We have been able to model not only key aspects of the cloud providers, like costs related to storage, availability, and

geographic location of the processing, but we can deduct them as well for the combined offering. The cryptographic technique of secret sharing splits data into n shares. Based on this cryptographic underpinning the combined cloud storage provider increases the availability above those of a single provider while additionally offering confidentiality for the data stored at each individual cloud provider assuming that only $k-1$ of them would collude. Currently the market is already too big and the combinations possible for selecting n different cloud storage offers and also keeping the cryptographic parameters underspecified, e.g. any k-out-of-n for $n \leq 15$ are overwhelming. Hence, manually selecting such multi-cloud storage solutions is impossible. We provide a first look into what one could do if one would model the internal workings of multi-cloud techniques and combine them with a market survey.

We show that not only does the secret sharing allow to uphold confidentiality, prevent from provider lock-in, and heightens the availability, e.g. to 99.9999%. It also reduces the cost one would pay for such a service. We show that a cost reduction as high as 50% can be achieved. As privacy and data protection are always of paramount importance in the cloud, we added a geographical limit to the calculation alongside the even quantum safe confidentiality increase from secret sharing: In our search of the market for combinations we always required that at least one provider from a specific geographic region, e.g. the EU or even a specific country, to enable reconstruction. This significantly eases the challenge to comply with policies and regulations, such as the GDPR [1], i.e. with the configuration of the target function used in this paper an adversary always needs to get at least one share from a storage provider within EU jurisdiction to reconstruct the data. Technically, the solution provides confidentiality protection, just like encryption, and the one share from an EU provider could be seen as 'a key' that stays within EU[17].

In the future we hope that this will foster more competition in the cloud provider market as we will have more easy ways to compare offers as well as to substitute one provider with another one. Therefore we have made our data base of cloud storage offers as well as the decision support system's target function available[18]. Finally, we hope that this allows to market a cryptographically increased security, i.e. secret sharing, for cloud storage as it even comes with additional cost savings.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 644962 (PRISMACLOUD) and additionally by Eureka member countries under Eurostars no 11450 (DRBD4Cloud).

[17] Whether or not those shares legally are still personal data is subject to an ongoing legal debate; our view follows the views of some scholars, e.g. [23] and in line with the analysis presented by the current European research project SODA https://www.soda-project.eu/wp-content/uploads/2018/02/SODA-D3.1-General-Legal-Aspects.pdf, accessed Aug. 2018

[18] https://github.com/Archistar/c3sp/blob/master/StorageProviders.json

References

[1] European Parliament and the Council of the European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal*, vol. OJ L 119 of 4.5.2016, pp. 1–88, May 2016.

[2] M. A. AlZain, E. Pardede, B. Soh, and J. A. Thom, "Cloud computing security: From single to multi-clouds," in *Proceedings of the Annual Hawaii International Conference on System Sciences*. IEEE, jan 2012, pp. 5490–5499. [Online]. Available: http://ieeexplore.ieee.org/document/6149560/

[3] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, nov 1979. [Online]. Available: http://portal.acm.org/citation.cfm?id=359168.359176

[4] H. Weatherspoon and J. Kubiatowicz, "Erasure Coding Vs. Replication: A Quantitative Comparison," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK, UK: Springer-Verlag, 2002, pp. 328–338. [Online]. Available: http://dl.acm.org/citation.cfm?id=646334.687814

[5] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the ACM*, vol. 36, no. 2, pp. 335–348, 1989. [Online]. Available: http://portal.acm.org/citation.cfm?doid=62044.62050

[6] M. Li, "On the Confidentiality of Information Dispersal Algorithms and Their Erasure Codes," *arXiv preprint arXiv:1206.4123*, pp. 1–4, jun 2012. [Online]. Available: http://arxiv.org/abs/1206.4123

[7] H. Krawczyk, "Secret sharing made short," *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, vol. 773, pp. 136–146, 1994. [Online]. Available: http://www.springerlink.com/index/0PDLWL2K4N952E2E.pdf

[8] A. Bishop, V. Pastro, R. Rajaraman, and D. Wichs, "Essentially optimal robust secret sharing with maximal corruptions," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9665. Springer, Berlin, Heidelberg, 2016, pp. 58–86. [Online]. Available: http://link.springer.com/10.1007/978-3-662-49890-3{_}3

[9] H. Krawczyk, "Distributed Fingerprints and Secure Information Dispersal," in *Proc 20th Annual ACM Symp on Principles of Distributed Computing*, IBM T. J. Watson Research Center. ACM, 1993, pp. 207–218.

[10] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS," in *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10*. New York, New York, USA: ACM Press, 2010, p. 229. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1807128.1807165

[11] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," in *16th ACM Conference on Computer and Communications Security, CCS '09*. ACM, 2009, pp. 187–198. [Online]. Available: http://eprint.iacr.org/2008/489.pdf

[12] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud-of-Clouds," in *Proceedings of the sixth conference on Computer systems - EuroSys '11*. New York, New York, USA: ACM Press, 2011, p. 31. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1966445.1966449

[13] T. Loruenser, A. Happe, and D. Slamanig, "ARCHISTAR: Towards Secure and Robust Cloud Based Data Sharing," in *Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on*, nov 2015, pp. 371–378. [Online]. Available: https://doi.org/10.1109/CloudCom.2015.71

[14] T. Lorünser, D. Slamanig, T. Länger, and H. C. Pöhls, "PRISMACLOUD Tools: A Cryptographic Toolbox for Increasing Security in Cloud Services," in *11th International Conference on Availability, Reliability and Security, ARES 2016, Salzburg, Austria, August 31 - September 2, 2016*. IEEE Computer Society, 2016, pp. 733–741. [Online]. Available: http://dx.doi.org/10.1109/ARES.2016.62

[15] T. Lorünser, H. C. Pöhls, L. Sell, and T. Laenger, "CryptSDLC: Embedding Cryptographic Engineering into Secure Software Development Lifecycle," in *Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018*, 2018, pp. 4:1—-4:9. [Online]. Available: http://doi.acm.org/10.1145/3230833.3233765

[16] A. Hudic, M. Tauber, T. Lorunser, M. Krotsiani, G. Spanoudakis, A. Mauthe, and E. R. Weippl, "A Multi-layer and MultiTenant Cloud Assurance Evaluation Methodology," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, 2014, pp. 386–393.

[17] M. Naldi and L. Mastroeni, "Cloud storage pricing," in *Proceedings of the 2013 international workshop on Hot topics in cloud services - HotTopiCS '13*. New York, New York, USA: ACM Press, 2013, p. 27. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2462307.2462315

[18] ISO, "ISO/IEC 19086-1: Information technology — Cloud computing — Service level agreement (SLA) framework and technology — Part 1: Overview and concepts," International Organization for Standardization, International Electrotechnical Commission, Standard, 2016, final Draft.

[19] ——, "ISO/IEC 19086-4: Information technology — Cloud computing — Service level agreement (SLA) framework — Part 4: Components of Security and of Protection of PII," International Organization for Standardization, International Electrotechnical Commission, Standard, Jun. 2017, committee Draft.

[20] A. Happe, F. Wohner, and T. Lorünser, "The Archistar Secret-Sharing Backup Proxy," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES '17. New York, NY, USA: ACM, 2017, pp. 88:1—-88:8. [Online]. Available: http://doi.acm.org/10.1145/3098954.3104055

[21] D. Demirel, S. Krenn, T. Lorünser, and G. Traverso, "Efficient and Privacy Preserving Third Party Auditing for a Distributed Storage System," in *11th International Conference on Availability, Reliability and Security, {ARES} 2016, Salzburg, Austria, August 31 - September 2, 2016*. {IEEE} Computer Society, 2016, pp. 88–97. [Online]. Available: http://dx.doi.org/10.1109/ARES.2016.88

[22] J. Stangl, T. Lorunser, and S. M. Pudukotai Dinakarrao, "A fast and resource efficient FPGA implementation of secret sharing for storage applications," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, mar 2018, pp. 654–659. [Online]. Available: http://ieeexplore.ieee.org/document/8342091/

[23] W. K. Hon, C. Millard, J. Singh, I. Walden, and J. Crowcroft, "Policy, legal and regulatory implications of a europe-only cloud," *International Journal of Law and Information Technology*, vol. 24, no. 3, pp. 251–278, 2016.