Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on , pp.1-5, Feb. 28 2011-March 3 2011
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5940844&isnumber=5940805

A Farewell to Trust: An Approach to Confidentiality Control in the Cloud

Martin Gilje Jaatun*, Åsmund Ahlmann Nyre*, Stian Alapnes† and Gansen Zhao‡
*SINTEF ICT, Norway
Email: {martin.g.jaatun,aasmund.a.nyre}@sintef.no
†Telenor Corporate Development, Norway
Email: stian.alapnes@telenor.com
‡South China Normal University, China
Email: gzhao@scnu.edu.cn

***Abstract*—This paper applies a divide-and-conquer approach to achieve confidentiality control in Cloud Computing. We sketch how a Redundant Array of Independent Net-storages (RAIN) for Cloud Computing can be designed using techniques originally intended for other purposes. The RAIN approach splits data into segments and distributes segments onto multiple providers. By keeping the relationships between the distributed segments private, the original data cannot be re-assembled. Further, with each segment small enough, each segment discloses no meaningful information to others. Hence RAIN is able to ensure the confidentiality of data stored on clouds.**

I. Introduction

Wireless technologies have enabled truly mobile computing, and a large part of the pending increase in mobile data can be attributed to cloud computing [1], since complex operations can be performed in the cloud while accessing the results via simple wireless devices. Security concerns are frequently cited [2] as one of the major obstacles to cloud computing adoption. In a traditional outsourcing scenario, technical and organizational security mechanisms contribute to protect a customer's data, but the most important factor is that the customer establishes a trust relationship with the provider (see Fig. 1). This implies that the customer acknowledges that if the provider is evil, the customer's data may be used improperly [3].

One aspect of Cloud Computing can be described as "outsourcing on steroids"; where both storage and processing is handled by one or several external providers, and where the provider(s) may be in a different jurisdiction than the customer. Not knowing where your data is physically located may be uncomfortable to the customer, and personal data may even be illegal to export from some jurisdictions [4]. Just like with traditional offshoring, settling disputes is more challenging when the provider may be on a different continent, which is all the more reason to limit the degree to which the customer has to trust the provider. This is the "need to know" principle in a nutshell - if the provider does not need to read the information, why should it be allowed to?

In this paper, we will describe a path toward a Cloud Computing scenario where the dependency on *trust* will be reduced through a divide-and-conquer approach, where each actor gets access to sufficiently small units of data so as to minimize confidentiality concerns[1]. In a way, our approach is the opposite of the aggregation problem in database security – we *de-aggregate* the sensitive data.

The remainder of the paper is structured as follows: In Section II we identify problem statements, and in Section III we outline the background for our contribution. In Section IV we sketch our solution, and discuss our contribution in Section V. We outline further work in Section VI, and offer our conclusions in Section VII.

II. Problem Statements

Cloud computing provides on-demand services to clients, relieving the clients of the burden of deployment and management of their own IT infrastructures and applications. The clients need only to choose the right providers for the needed infrastructures and applications. The services are provided in an off-premises manner and delivered via the Internet. This pattern for IT capacity provisioning is appealing in most cases due to its characteristics such as convenience, rapid deployment, cost-efficiency, and so on. However, when relying on off-premise services for data storage, clients have the common security concerns:

- **Data availability.** With cloud computing, data are kept and managed by cloud storage providers at remote sites. When keeping data at remote systems owned by others, data owners may suffer from system failures of the service provider, as system failures will mean that data will become unavailable if the data depends on a single service provider. As no cloud service provider can guarantee 100% availability of services, the data kept and managed on a cloud will suffer data unavailability when the provider is out of operation. Unavailability of data could be a disaster to some business, especially to those who heavily reply on the data for business transaction processing.
- **Data Confidentiality.** As data are kept and managed by cloud storage providers, there is no way for data owners

[1] Note that this approach to confidentiality may not be acceptable in certain high-security environments, such as classified military installations – but then again, it is unlikely that these environments will be employing public cloud computing approaches in the foreseeable future anyway.

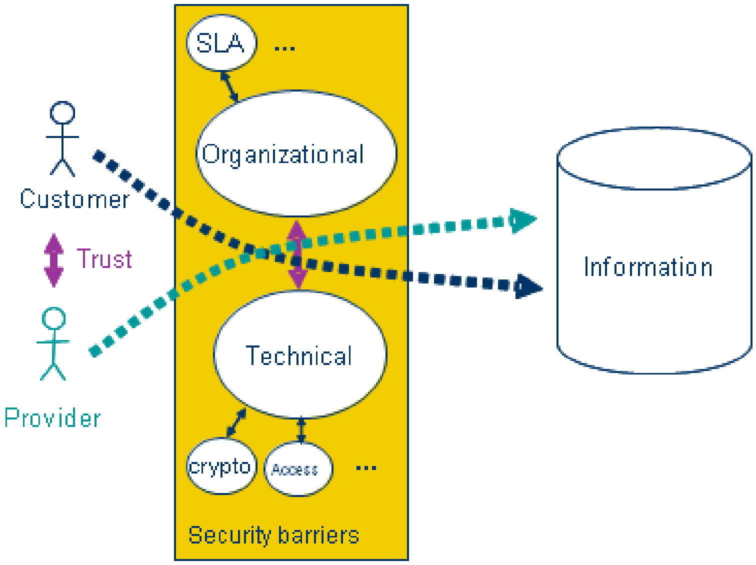


Fig. 1: Outsourcing and trust

to prevent the providers to allow others to access the data. It is obvious that data owners will not tolerate their data being accessed without their authorization. Cloud storage providers could enforce access control on the modifying of data. But the access control infrastructure is still potentially under the complete control of the providers, the providers can still override the access policies imposed by the data owners.

- **Data Integrity.** Similar to the issue of data integrity, data owners have no way to prevent their data being tampered with due to the complete control power held by the cloud storage providers.

III. Background

In previous work [2], we identified five deployment models of cloud services designed to ease users' security concerns:

- **The Separation Model** separates storage of data from processing of data, at different providers.
- **The Availability Model** ensures that there are at least two providers for each of the data storage and processing tasks, and defines a replication service to ensure that the data stored at the various storage providers remains consistent at all times.
- **The Migration Model** defines a cloud data migration service to migrate data from on storage provider to another.
- **The Tunnel Model** defines a data tunneling service between a data processing service and a data storage service, introducing a layer of separation where a data processing service is oblivious of the location (or even identity) of a data storage service.
- **The Cryptography Model** extends the tunnel model by encrypting the content to be sent to the storage provider, thus ensuring that the stored data is not intelligible to the storage provider.

By use of these deployment models, we have shown that through duplication and separation of duty, we can alleviate availability and integrity concerns, and to some extent also confidentiality, by implementing encrypted storage. However, even with encrypted storage, we still have to trust the encryption provider with *all* our data.

The main motivation for confidentiality control in the cloud is currently various privacy-related legislation forbidding the export of sensitive data out of a given jurisdiction, e.g. the Privacy legislation in the EU [4]. The current solution to this problem has been to sidestep it, by offering geolocalized cloud services, where a customer may request the cloud provider to ensure that the sensitive data is only stored and processed on systems that are physically located in a geographically defined area, e.g., within the borders of the European Union. However, this is rapidly becoming a moot point, since cloud service providers typically run global operations, and although data might physically reside in one jurisdiction, it will in principle be accessible from anywhere in the world.

Although misappropriation of data by cloud providers have not been documented, Jensen et al. [5] show that current cloud implementations may be vulnerable to attack. Ristenpart et al. [6] demonstrate that even supposedly secret information such as where a given virtual machine is running may be inferred by an attacker, highlighting another attack path.

Krautheim [7] proposes to achieve cloud security through the introduction of Trusted Platform Modules (TPM) in all datacenter equipment. It is not clear, however, how the user could verify that a TPM is indeed present in any given cloud infrastructure. You might argue that the cloud provider could assert, and have an auditor confirm that they are using a TPM, but this is really not much better than today's situation where providers are asserting that they will treat your data properly, and all their certifications is a testament to them staying true to their words.

Mowbray et al. [8] describe a scheme for privacy in the cloud by using obfuscation of sensitive data. The examples they provide may not be applicable to general data, but we believe this represents a promising start that may yield more general results in the future.

Bogdanov et al. [9] have developed a framework [10] for distributed privacy-preserving computations, based on the principles of secure multiparty computations. They state that due to (among other things) the scaling problems of secret sharing protocols, only solutions with 3 to 5 nodes are practical. We avoid this problem in our solution, since we leave the data unencrypted.

IV. Approach

We extend the deployment models [2] with a new concept where we split up the data between several independent (non-colluding) storage providers in a Redundant Array of Independent Net-storages (RAIN), in such a manner that a single chunk does not compromise confidentiality. The data can then be stored using one or several cloud storage providers (probably duplicated, according to the deployment models).

A. Enter the Bots

At this point, we are tempted to paraphrase Larry Norman[2]: Why should the Blackhats have all the good bots? In other words, we propose to organize the various elements in our distributed cloud architecture as a traditional multi-tier Command & Control (C&C) botnet, e.g. as described by Wang et al. [11].

We introduce a new type of cloud service provider which assumes the role of the botnet C&C node, and which is in charge of assembling the information and presenting it. This keeps all processing in the cloud, but leaves us with the problem that we have to trust this provider with our information. The resulting configuration is illustrated in Figure 2a.

The key property of the solution is that all the subnodes (cloud processing providers) and leaf nodes (cloud storage providers) only get to observe a small subset of a user's data, and that these nodes are prevented from associating a given piece of data with a specific user, or with other pieces of the same dataset. Ultimately, it will be like breaking open a large number of jigsaw-puzzles and distributing the pieces among storage providers – a single provider will not be able to determine where the pieces come from, or even if they are part of the same puzzle.

Referring to Figure 2a again, we would need to employ three different cloud cryptography providers for the tunnels from the C&C node to the subnodes. Of course, if the number of subnodes becomes large enough, it would be necessary to introduce additional tiers in the hierarchy, in order to minimize re-use of cryptography provider in any one node. We also assume that communication from the user to the C&C node is protected by, e.g., conventional TLS.

Note that we do not propose to make the cloud processing provider work with encrypted data; the confidentiality control is achieved through the de-aggregation of information, and hiding the relationships between the processing providers. Also note that assuming the volume of such "botnet computations" is large enough (i.e., many enough users employ this technique), it is also possible to re-use providers, since it will not be possible for a provider to relate two different processing tasks with each other.

For the truly paranoid, it could be possible to introduce uncertainties by routinely accessing bogus data, but although the user will know which data is real, and which is bogus, this will introduce the need for some "intelligence" on the client (to separate the wheat from the chaff), and we find ourselves transported to the alternative solution presented in Section IV-D.

B. Example

To illustrate the concept, we will in the following consider the storing of bitmap images in the cloud. Figure 3 shows a 177x216 grayscale bitmap image, with 16 bits per pixel. We can slice this image into 177 (vertical) pixel vectors, of 216 pixels (or 3456 bits) each. For our purposes, a single slice of the picture does not reveal useful information to the observer, and this information can be stored unencrypted as long as it is not possible to combine it with the other slices.

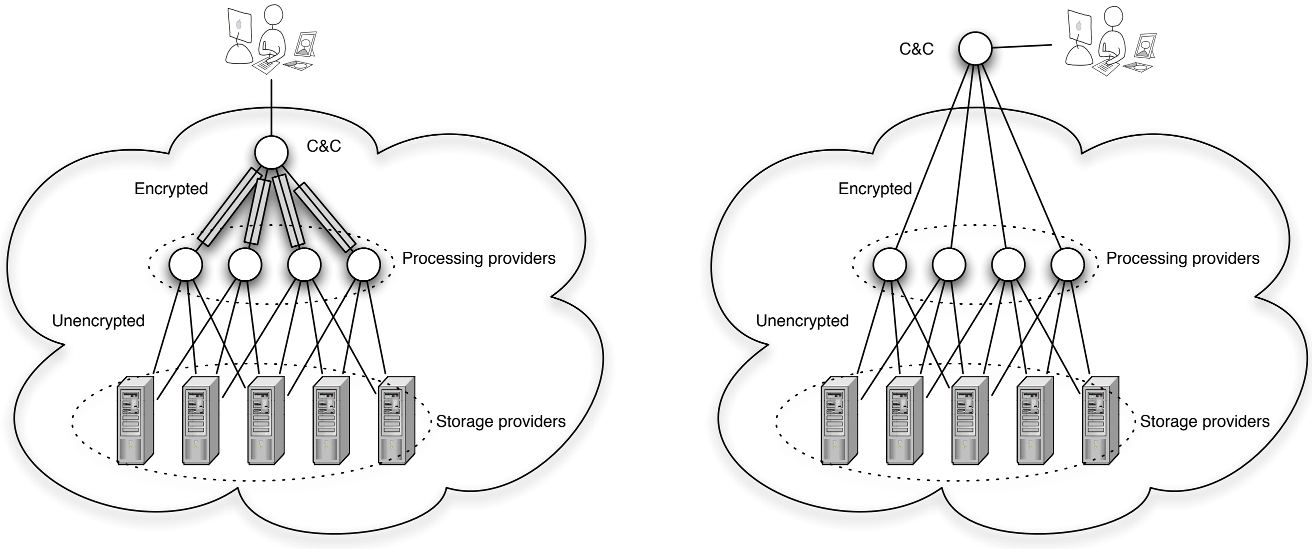
The C&C node performs the slicing of the image, and randomly distributes the slices among (say) 9 subnodes (each subnode gets 384 slices in this case), transferring the information to the subnodes using an encrypted tunnel. Each subnode then stores the slices independently using as many cloud storage providers as available (ideally one for each slice, but even for this small example we would probably be hard pressed to find 384 independent providers). To prevent observability, the subnodes may use an encrypted tunnel to transfer the data to the storage providers, but this may be sacrificed for the sake of efficiency.

It is the responsibility of the C&C node to keep track of which subnode has received which slices, but also to record the location where each slice has been stored – it is preferable to re-use subnodes, but if they should somehow disappear, the C&C node must be able to re-create the network for a new subnode.

When the image is to be retrieved, the user will instruct the C&C node to fetch all the slices, and any editing operations etc. can be performed. Note that when updates to the image needs to be stored, only slices with changes need to be rewritten.

Admittedly, this is a toy example, with all the real processing being performed by the C&C node – the real challenge comes when we need to perform complicated processing on

[2] Author of the song "Why Should the Devil Have All the Good Music?"(1972), covered by Cliff Richard and others.



(a) Pure Cloud Solution

(b) With Intelligent Client

Fig. 2: Divide-and-conquer Cloud Security

Fig. 3: 177x216 grayscale image

each subnode. This will also introduce the need for more sophisticated "slicing" of data.

C. Criteria

Since we perform the slicing and distribution of data in order to achieve data confidentiality, it is important that the slicing and distribution process adheres to the following criteria:

- Data must be sliced into segments small enough such that each segment bears no meaningful information to malicious entities[3]. With data sliced in this way, malicious entities may be able to access an individual data segment, but the access to the data segment should not compromise the confidentiality of the data as a whole.
- Data segments must be distributed in a random manner, such that it is not possible to establish the relationships between data segments without knowledge of the original data. The relationships between data segments are kept secret by the data owner.

With the above two criteria strictly enforced, the proposed approach would be able to ensure the confidentiality of data. This is achieved without encrypting the data.

D. Alternative Solution

If we are unwilling to trust the C&C provider described in the previous section, an alternative solution is place this functionality on the client, i.e. running on the user's own infrastructure, as illustrated in Figure 2b.

This solution requires a certain amount of computing power present on the client side in order to (re-)assemble all the different pieces of information produced by the cloud processing provides, and may thus not necessarily be considered a pure cloud solution.

V. Discussion

Cloud service providers have been identified as potential targets of attack simply because of the vast amounts of data they store on behalf of their multitude of customers. In this sense, it may be in the providers' best interest to "know less" - if even the provider cannot access the customers' data directly, there is little point in attacking them.

Strictly speaking, most users would probably be happy if it were possible to impose universal usage control [12] on data submitted to providers (a sort of "reverse DRM", where end-users get to control how multi-national corporations use their data), but despite Krautheim's efforts [7], we do not believe this will be a reality in the foreseeable future. Thus, it would seem that the easiest way to control what a provider does with your information is to hide it - either through encryption

[3] By entities we here primarily mean cloud providers, but also possibly other users/customers of these cloud providers.

(as previously proposed for the storage providers) or through separation.

We have specified the use of multiple cryptography providers as well as storage and processing providers, and it is necessary to prevent "rotation" of providers (avoid using same provider to encrypt different versions of same data), as one might otherwise risk all providers having all data after a while. It is thus better to accept that each provider has partial, albeit updated, information. However, due to the botnet-like hierarchy, the providers do not know to whom the information belongs, and the value of the information is practically nil.

In a real-life setting, there will be cases where very small units of data carry a significant amount of sensitive information, such as blood type for patients. It will thus be imperative that not only shall it not be possible to match e.g. a blood type to an identity, but in storage it should also not be possible to determine what the data item refers to.

Since we place all our trust in the C&C node, it will remain as a "single point of trust" as long as it is realized as part of the cloud. It would have been desirable to strengthen this by ensuring that the C&C node provider only sees the information as we see it ourselves, preventing it from mining stored information. However, as long as the C&C node is required to keep track of all the data items (or slices, as in the example), there is nothing to prevent it from accessing this information as it pleases. Currently, only the alternative solution in Figure 2b keeps this information out of the cloud.

VI. Further work

This divide-and-conquer approach may be suitable for some applications, but the ultimate holy grail is absolute confidentiality in the cloud, and thus a deliverance from trust. Only then can cloud computing deliver on the dream of computing power as a utility akin to power, water and gas.

As a first step, we will implement a prototype of our divide-and-conquer approach, specifically to gauge performance impacts on typical cloud applications. One particular challenge in this respect is to determine the optimal slicing strategy for arbitrary data. It is likely that a trade-off between security and efficiency will have to be made in order to capitalize on the advantages of the Cloud Computing paradigm. The prototype will be targeted toward a "sensitive but unclassified" application, representing a realistic use case.

Furthermore, we intend to examine practical experiences from projects such as Free Haven [13] and OceanStore [14] to evaluate how our solution would scale.

VII. Conclusion

We have described an idea on how to achieve confidentiality in the cloud through dividing data in sufficiently small chunks. We believe that this may be useful for selected use cases, but experimentation and practical experience will be necessary to validate the approach.

Acknowledgements

This work has been supported by Telenor through the SINTEF-Telenor research agreement and by China State Key Lab of Software Engineering through SKLSE2010-08-22.

References

[1] S. Cherry, "Forecast for cloud computing: Up, up, and away," *Spectrum, IEEE*, vol. 46, no. 10, pp. 68 –68, Oct. 2009.

[2] G. Zhao, C. Rong, M. G. Jaatun, and F. Sandnes, "Reference deployment models for eliminating user concerns on cloud security," *The Journal of Supercomputing*, pp. 1–16, 2010, 10.1007/s11227-010-0460-9. [Online]. Available: http://dx.doi.org/10.1007/s11227-010-0460-9

[3] Å. A. Nyre and M. G. Jaatun, "A Probabilistic Approach to Information Control," *Journal of Internet Technology*, vol. 11, no. 3, pp. 407–416, 2010.

[4] E. Parliament., "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." pp. 31–50, 1995.

[5] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On Technical Security Issues in Cloud Computing," *Cloud Computing, IEEE International Conference on*, vol. 0, pp. 109–116, 2009.

[6] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 199–212.

[7] F. Krautheim, "Private virtual infrastructure for cloud computing," in *proceedings of the Workshop on Hot Topics in Cloud Computing, HotCloud*, 2009.

[8] M. Mowbray, S. Pearson, and Y. Shen, "Enhancing privacy in cloud computing via policy-based obfuscation," *The Journal of Supercomputing*, pp. 1–25, 2010, 10.1007/s11227-010-0425-z. [Online]. Available: http://dx.doi.org/10.1007/s11227-010-0425-z

[9] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: a framework for fast privacy-preserving computations," Cryptology ePrint Archive, Report 2008/289, 2008, http://eprint.iacr.org/.

[10] "Cybernetica news blog - sharemind," http://research.cyber.ee/sharemind/, visited: Sept. 9, 2010. [Online]. Available: http://research.cyber.ee/sharemind/

[11] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "A systematic study on peer-to-peer botnets," in *ICCCN '09: Proceedings of the 2009 Proceedings of 18th International Conference on Computer Communications and Networks*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–8.

[12] J. Park and R. Sandhu, "The UCON_ABC usage control model," *ACM Trans. Inf. Syst. Secur.*, vol. 7, no. 1, pp. 128–174, 2004.

[13] R. Dingledine, M. J. Freedman, and D. Molnar, "The free haven project: Distributed anonymous storage service," in *Proceedings of the Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[14] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz, "Pond: the OceanStore Prototype," in *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST '03)*, 2003.