International Journal of Pure and Applied Mathematics
Volume 85 No. 1 2013, 1-11
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: http://dx.doi.org/10.12732/ijpam.v85i1.1

CLOUD CRYPTOGRAPHY

Kelsey Rauber

Mathematics Department
New York City College of Technology
300, Jay Street, Brooklyn, NY 11201, USA

Abstract: As kids we often wonder what it would feel like to touch a cloud, play on it, maybe even build a living environment on it. We now have found a way to step into it: cloud computing! To a kid, this statement might evoke false imagery, but to scientists all over the world, we are a step closer to making the impossible possible. Cloud computing is what we are talking about. What will it offer, how will we use it? How is it being used right now?

This paper focuses on the key element with whose absent the entire system could fall apart: Security! What are the main components that need to be secured? How will cloud computing revolutionize our computer experience? Where do Security-As-A-Service, Homomorphic Encryption and Functional Encryption fit in? How do they work and how will they keep our information secure?

AMS Subject Classification: 94
Key Words: cloud cryptography

1. What is Cloud Computing?

Anyone interested in telephony or Internet setups can easily imagine why it is called cloud computing. The word or origin of the word "cloud computing" comes from the diagram drawing of a network, when sybolizing a mutual master group. The idea of cloud computing originally came up in the 1960s when John McCarthy mentioned the future of computers would be in a public utility space.

Received: March 30, 2012

It was further elaborated on in Douglas Parhil's book "The Challenge of the Computer Utility".

The idea is that the user no longer needs to use Servers and expensive cooling and storing devices. All they need is access to the Internet and the cloud. This way upgrades and software license and license agreements can be omitted. No end-user knowledge is necessary, which makes it easily accessible to any web-savvy individual. The software would turn into service ("Software as a Service" or "SaaS") rather than a product.

The architecture of a cloud is based on loose coupling, which is comparable with a messaging queue.

The first company to have made such a cloud become popular is Amazon. In 2006 they approached the utility computing after the dot-com bubble and started the Elastic Compute Cloud (aka EC2).

In 2008, Eucaluptus offered an open source cloud for private usage, which was followed by Open Nebula in the same year, which offered private, as well as hybrid clouds (more information on this follows), see [1].

2. There are Three Different Types of Clouds

- Public clouds: They contain web applications for the public
- Community clouds: A group of people with common goals and concerns would access this cloud
- Hybrid clouds: This would combine more than one cloud, which would offer a wider variety on writing applications and using them in combinations.

For the future we are expecting an interncloud, cloud of clouds, a data system that would contain every bit of information, overseeing all clouds and information, making any kind of storing devices obsolete. Imagine that. The step after is to simply implement that intercloud into our brains (this conclusion is not based on any academic paper). [2].

The last technical aspect I'd like to elaborate on is the set up of a cloud. There are three layers, which would eventually lead to the pricing of using a cloud.

- Infrastructure (borrow a server)
- Platform ("Platform as a Service" or "PaaS")

- Software ("Software as a Service" or "SaaS")

2.1. Best Case Scenario

In an ideal world, clouds would be used as a sharing tool to optimize applications. It would speed up scientific publishing, because research would be globalized and the amount of double tracking processed would hopefully be decreasing.

Clouds would offer a whole new perspective on a way to share medical information, decreasing errors. Patient information, doctors, observations and cures would easily be shared through this quick and efficient cloud.

It would lead to a greener, more energy efficient way of computing.

2.2. Worst Case Scenario

In some extreme cases, programmers have been discussing the possibility of a digital Pearl Harbor. This would be caused by hackers who would take down our system rapidly, leaving little to no memory at all. Imagine all of your stored input on all of your electronic devices - gone. It is in fact comparable with what was expected at the millienium - did not happen. But most likely the biggest issues of the cloud security will be based on secrecy and monetary concerns.

3. The Types of Data

One of the biggest privacy issues we are facing right now is the guarantee of privacy while working on an "SaaS". While working on open files (decrypted), haking becomes an easy game. To ensure safety in clouds, a good understand of the different stages of data transmission is required.[3].

- Data in transit
- Data at rest
- Processing of data (including multitencancy)
- Data lineage
- Data provenance
- And data remanence

The first three items seem trivial. Using the example of an email, one coud say, data in trasit is sending and email, data at rest is the email in your mailbox and processing of data refers to typing up a response. To elaborate, the description of the remaining types follow:

3.1. Multitenancy

It is a software architecture where multiple clients are running off of one server.

3.2. Data Lineage

Data lineage, as the word already describes, includes the data history and is stored in data warehouses. Nowadays we need to be able to track alterations to data to keep businesses satisfied. A simple example of data lineage is Mac's Time Machine.[4].

3.3. Data Provenance

Data provenance is the information that helps determine the derivation history of a data product, starting from its original sources. It is hard for a computer to check the "correctness" of this information. These are the applications of data provenance: [5].

- Data Quality: Lineage can be used to estimate data quality and data reliability based on the source data and transformations [5]. It can also provide proof statements on data derivation [5].
- Audit Trail: Provenance can be used to trace the audit trail of data [5], determine resource usage [5], and detect errors in data generation [5].
- Replication Recipes: Detailed provenance information can allow repetition of data derivation, help maintain its currency [5], and be a recipe for replication [5].
- Attribution: Pedigree can establish the copyright and ownership of data, enable its citation [5], and determine liability in case of erroneous data.
- Informational: A generic use of lineage is to query based on lineage metadata for data discovery. It can also be browsed to provide a context to interpret data.

For example, if we consider financial information and the exchange rate of Dollars to Swiss Francs. To get the correct answer, you will have to determine firstly which Dollar is requested. If the computer is not set up a way to question this transaction, it will not recognize the provenance.

3.4. Data Remanence

The best description of data remanence for a starting computer user can be found on Zapthink.com [6]: "Everybody knows that dragging a file into the trash and then emptying the trash doesn't actually erase the file. It simply indicates to the file system that the file is deleted, but the data in the file remain on the hard drive until the file system eventually overwrites the file. If you require the actual erasure of deleted files, then you must take an active step to erase the portion of the drive that contained the file, perhaps by explicitly overwriting each bit of the original file. Even then, it may be possible (although generally quite difficult) to recover parts of the original file, due to the magnetic properties of the storage medium. We call this problem data remanence."

We question now, what is the best way to secure all of these data types and transformation during cloud usage?

4. Security-As-a-(Cloud) Service

Security-As-a-Cloud Service would be a subscription based opportunity to secure the online data as it is being received or transferred. According to the book cloud Security and Privacy [7] this would not only be cost efficient but also use our resources in a much more proficient way. The endpoint user would not have to deal with viruses because it would already have been in effect by the time the data is received. Mobile phones and other apps would profit because their capacity isn't high compared to other devices as well as, if the comparable capacity doesn't supports the encryption programs. A good example to use here is email. Any email provider nowadays implements a spam detector. However, security and discretion remain with the customer. Users still have the option to see and open mail that was posted in a spam mailbox. (see postini) For web-content filtering the anti-virus program reads the HTTP coding and decides if it is harmful or not. The endpoints of all online searches belong to the organizations. This is where the cloud must be on top of, if not excell at the Identity Management as a Service! But how?

5. Identity Management as a Service

Identity management-as-a-Service (IDaaS) has recently emerged and there are many aspects of IDaaS that are deficient. The most critical problem is authentication. Looking at the Internet and where how it operates today, you see that online banking, email etc. already implement a way of authenticating their users. However, there are more points that need to be covered in cloud computing to secure efficiency as much as safety. One of the most significant problems for CSPs (Certified Safety Professionals) concerning IDaaS providers, and "developing some form of collaborative meta system." [11] Any business using cloud computing will face issues of turn-over of employees, freelance workers etc.; namely, outside forces who were given access to the cloud, but lose their privileges. There are three processes to Identity and access management, they include, authentication, authorization and auditing. Auditing is the process of keeping track of in- and outgoing personnel.

6. Cryptographic Cloud Storage

The concerns that surround cloud computing run the gamut. Some of the biggest questions on how to safely run and store information is currently being researched by digital companies and universities all around the world. How do we prove the integrity of a user without having to download any software? In S. Kamara, K. Lauter's paper "Cryptographis Cloud Storage" they propose a cryptographic storage device that would be run similarly to a PKI (public key infrastructure), but with a slightly more complex approach.[8]. The objective is to keep track of

- Confidentiality; meaning, the storage provider should have no knowledge of the users information. Integrity; If there are changes to the information by an outside source it will can be detected
- Availabilty
- Reliability
- Efficient retrieval
- Data sharing

They're starting point is the idea, that information must be encrypted before sending it off to the storage provider. This works on a single writer/single

reader principle, but how do you share information? To have a fully functional cryptographic cloud storage system the following three checkpoints would need to be in action:

1. Data processor (DP)
2. Data Verifier (DV)
3. Token Generator (TG)

Let's say Amy is working on a couple of papers concerning pet care. She writes one on cats, one on dogs and one on birds. She owns a private cloud and stores her papers with a storage provider. The first time she uses the cloud storage a cryptographic key is provided, also known as a master key. It is stored on Amy's system and no one else has this key, not even the storage provider. After Amy finishes writing her paper, she uploads it to the data processor (DP). The data processor then encrypts and encodes it, after attaching metadata (such as time, size, keywords, etc.). Now it is safely stored with the storage provider, encrypted, so that no one can touch the information. If Amy needs to verify the information, she uses the Data Verifier, through which she ascertains the integrity. To retrieve the paper she uses the Token Generator (TG) . The Token Generator produces a token, which then is sent to the storage provider. The storage provider can find the appropriate encrypted data and sends it back to Amy. Amy uses her master key to decrypt the paper she received. If Amy decides that she would like to collaborate on one of the papers she can do so. She uses Bob, a dog specialist to proofread her paper on dog care. She doesn't want him to see the other two papers, since they are none of his concern. Therefore, Amy creates a token for the dog paper and only the dog paper, along with a credential to give him access. She sends the token to Bob, who then sends it to the storage provider. The storage provider sends him the encrypted document, which Bob decrypts with his credentials. For this to work smoothly, and for maximum security all of the client applications must be either open-source or made by someone other than the storage provider.

The keyword search would work in a similar manner. A token would be generated for the keyword and the provider would send back the information without encrypting the document.

7. Functional Encryption

Brent Waters from the University of Texas in Austin is currently working on this dilemma of safety of cloud computing with complicated authorizations. He

was elected one of the Microsoft Research Faculty Fellows of 2011. In his research he has recognized that the traditional encryption that we have today does not comply with many cloud systems and it is not efficient in authorizing users. His Functional Encryption focuses on embedding certain access predicates directly into the ciphertext. The predicates would be attributed in a way that would only allow the user to access the data he/she was meant for.[9]. This attribute-based encryption would follow a dual system encryption. In this method, Waters found a way to overcome partitioning. In a dual encryption system the keys and ciphertexts are either normal or semi-functional. The normal key decrypts either one, normal or semi-functional. The semi-functional key only decrypts the normal ciphertext. The semi-functional key and ciphertext are only applied when proof of security is needed. The proof utilizes a hybrid argument applying games:

- First game is the real security game: It checks for normal keys and ciphertext
- Second game checks for semi-functional ciphertext which is unlocked with the normal key.
- Following games are changed to semi-functional keys one by one.
- By the final game all keys are useless.

There is a catch: If the key and ciphertext have the same tag, the decyption will fail, regardless of semi-functionality. These tags need to be replaced by nominally semi-functional keys, which are able to decrypt semi-functional ciphertexts.

8. Public-Key Cryptography

Public key encryption is an essential method that has served as an important corner stone to cryptography. Even as it is becoming outdated, the knowledge of how it works remains important. As a broad example, I will use sending an email. The public key would be your email address for example. The general public has access to send you an email, however they cannot read your email. Once they have written up an email, they encrypt it with your public key (your email address) and send it to you. After this has happened, they can't undo the message, since they don't know how to unlock your lock, they only know how to encrypt the message with your public key. To read the message you

must log into your email account and you can then read the message sent to you, since it has been decrypted.

9. Homomorphic Encryption

To keep data encrypted but searchable at the same time there has been a new breakthrough in cryptology known as homomorphic encryption. It was developed by Craig Gentry (IBM).[10]. Gentry's first paper [11]. was based on ideal lattices. Taken from Craig Stuntz's Weblog [12]., the second paper based on Marten van Dijk, Craig Gentry, Shai Halevi and Vinod Waikuntanathan's paper can be broken down into pretty basic arithmetic operation.

- Integer Division
- Modular Arithmetic
- Modulo 2 Arithmetic and Binary Operations

10. Programs as Digital Circuits

This is where the homormorphic encryption has made a breakthrough. This encryption form operates the functions encrypt, decrypt as well as KeyGen (A license or product key generator, like the token discussed in the previous section) and in addition also Evaluate, wherefore the Boolean function is used.

Stuntz uses the following example (as a C program example):
bool $a = b?c : d$(where a, b, c, d are all bools) This means if (b) $a = c$, else $a = d$. This would be impossible to solve for a program where b was encrypted. If b can be written as binary operations the statement can be rewritten as
bool $a = (b\&c)|((!b)\&d)$
Which reads as $a = b$ AND c OR not b and d.

The homomorphic encryption equivalent:

CypherText $a = (b\ h_{\text{and}}\ c)\ h_{\text{or}}\ (h_{\text{not}}(b)\ h_{\text{and}}\ d)$
Where h_* operators are the homomorphic equivalents of usual Boolean operations. The catch to this encryption method is such, that the length of the plaintext must be known.

11. Conclusion

As of right now, it does not seem like there aren't enough opportunities to secure our information in a cloud. The biggest issues will be sharing data and keeping an eye on the "correctness" of the information. We will have to wait and see if these methods of security (Homomorphic encryption, functional encryption) will turn out to be feasible and work for everyone involved. Right now it seems to massive to be time efficient and too unexplored to provide enough experience to make an educated decision. Nonetheless, there are many ideas out there and as long as only a few people use the cloud and secure it with the Security-as-a-Service, the data should be pretty secure.

Acknowledgement

I would like to thank Dean Brown from School of Arts and Sciences for providing me with NSF grant to work on this project over summer 2011. I truly appreciate the work my mentor and supervisor, Professor Kahrobaei did for me, suggesting this interesting project and her ability to advised me through it.

References

[1] Collection of paper, Crypto cloud Computing Microsoft Research, *Turning Ideas into Reality*. August 1, 2011. http://research.microsoft.com/en-us/projects/cryptocloud/

[2] *Cloud Computing*, Wikipedia, the Free Encyclopedia. Web. 19 Aug. 2011, http://en.wikipedia.org/wiki/cloud-computing/

[3] R. L. Krutz, D.V. Russell, *cloud Security a Comprehensive Guide to Secure cloud Computing*. Indianapolis, IN: Wiley Pub., 2010.

[4] *Data Lineage: The Next Generation*, The Data Administration Newsletter TDAN.com. 1 Aug. 2011. http://www.tdan.com/view-articles/8151

[5] *A Survey of Data Provenance Techniques*, Y.L. Simmhan, B. Plale, D. Gannon.

[6] *"Data Remanence: Cloud Computing Shell Game*, ZapThink. 3 Aug. 2011. http://www.zapthink.com/2011/05/19/data-remanence-cloud-computing-shell-game/.

[7] T.Mather, S. Kumaraswamy, and L. Shahed. *cloud Security and Privacy: [an Enterprise Perspective on Risks and Compliance].* Sebastopol, Calif. u.a.: O'Reilly, 2009.

[8] *Cryptographis cloud Storage.* FC 2010 14th Financial Cryptograpy and Data Security International Conference (2010 01 25 - 2010 01 28 Tenerife, ESP); Kamara S.; Lauter

[9] *Encrypting Data in the cloud Brings Win for Texas.* N. Zeitler. Reviews and News on Tech Products, Software and Downloads — PCWorld. 19 Aug. 2011. http://www.pcworld.com/article/237822/encrypting-data-in-the-cloud-brings-win-for-texas.html.

[10] E. Naone. "Homomorphic Encryption - Technology Review." Technology Review: The Authority on the Future of Technology. Web. 19 Aug. 2011. ¡http://www.technologyreview.com/computing/37197/.

[11] C. Gentry, *Fully homomorphic encryption using ideal lattices*, Symposium on the Theory of Computing (STOC), 2009, pp. 169-178.

[12] C. Stuntz, *A Math Primer for Gentry's Fully Homomorphic Encryption*, http://blogs.teamb.com/craigstuntz/2010/04/08/38577/.

[13] V. Lyubashevsk, *Ideal Lattices*, http://people.csail.mit.edu; web: http://people.csail.mit.edu/joanne/idealtutorial.pdf.