TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Institute of Informatics
Chair of Information Systems

Analysis of Digital Integrity: Cloud Service Provider Case

Master's Thesis

Student: Oskar Poola
Student code: 132245IVGM
Supervisor: prof. Ahto Buldas

Tallinn 2015

I hereby declare that this thesis, which is the result of my work as an independent, are presented in Tallinn University of Technology to apply for a Master's degree, and on that basis have not previously applied for an academic degree.

Author Oskar Poola "....."..... 2015

The work complies with the applicable requirements.

Instructor Dr. Ahto Buldas "....."..... 2015

Approved by "....."..... 2015

Abstract

Digital integrity in an adjective that technological solutions can create for a system. Integrity is not as understandable as we think it to be in different cases and systems. Different scenarios will be shown and that every situation is different in terms of digital integrity. This work aims to bring to light what integrity means in a certain case such a data owner storing data in a third party cloud storage service provider. The author will provide different models to show, what can be the most effective protocol for cloud storage in an insecure world with potentially malicious actors.

Abbreviations	
SLA	Service Level Agreement
NSA	National Security Agency
SHA	Secure hash algorithm
CA	Certification Authority
TSP	Trust Service Provider
ISP	Internet Service Provider
ISKE	Estonian Security Management Standard
STORK	Secure idenTity acrOss boRders linKed
EPSOS	European Patients Smart Open Services
ECODEX	Enabling access to justice systems across Europe
NDA	Non-Disclosure Agreement
SPOCS	Simple Procedures Online for Cross Border Services
PKI	Public Key Infrastructure
eID	Electronic Identification
eIDAS	Electronic identification and trust services
DARPA	Defense Advanced Research Program and Applications
TSA	Time Stamp Authority
ECA	Estonian Certification Authority
$\mathrm{Sig}_C\{x\}$	Data x signed using a private key of C
$h(x)$	hash of x

List of Figures

Contents

1 Introduction

This work presents a theoretical analysis about the meaning of integrity and how integrity is achieved in practical scenarios. It turns out that the definitions of integrity in todays official documents like standards are quite misleading. They do not provide a definitive characteristic to data integrity. The author describes technical primitives that can be used to achieve data integrity, the use of a single technology is not the answer. Several sequenced models are defined. Each model is explained through normal verification and disputes to show that each model has its drawbacks until a working model is presented. The cloud case solution is an effective data protocol that takes into account checking, verification and disputes.

The motivation to analyse this certain topic began at an internship at Guardtime AS, when the lack of integrity awareness was discovered. The topic is relevant in todays world, when talking about security of electronic data in any form. The general misconception is that, when the security of a digital object is discussed, usually the accessibility and confidentiality of the system is taken into strong consideration, but what gives information its value is actually the integrity or the trustworthiness over an extended period of time. The main goal of the work is to analyse digital integrity and how they would stand up against scrutiny, when new technologies are emerging on the market that can be used more efficiently to secure digital data. The expected outcome of the study is that the current security solutions are becoming redundant and ineffective in the face of new emerging cryptographic technologies that ensure integrity of digital data. This work can create benefits for information intensive sectors by rethinking the basic concepts of information integrity. Different definitions of integrity are investigated in this work. Cryptographic and non-cryptographic solutions available today are exam-

ined and how in certain case studies these technologies would have played a positive outcome in their use. The author examines Estonia's digital signature process and how it gives electronic data it's integrity. The CIA triangle (Confidentiality, Integrity and Availability), defines most security systems for users today. The focus is however mostly on confidentiality and availability, an argument will be made for integrity and how the technology in todays market can make a enormous difference at how electronic data security is perceived.

Current digital trust services provided and how efficient of a mechanism it is in the current digital era. Estonia's digital signature and how they have managed based on the European Union Directive 1999/93/EC "Community Framework for Electronic signatures" which defines the requirements for digital signatures certification providers and also ISKE[11], which is the Information Systems Three Tier Baseline Security Standard.

Such solutions as error correction codes and error detection codes, which are non-cryptographic solutions will be examined and along with hash functions, message authentication codes, time stamps and digital signatures of which the latter are cryptographic codes are examined.

Integrity of any electronic data is relevant in todays world, when we are talking about security of electronic data. In any case it can be records, data, information etc. Any information that is in a electronic form. The implications of an effective solution for signing digital data are impressive. As mentioned before the amount of digital data in todays world is rapidly increasing and the integrity of data is not as often under question as it should be. It is not only objects that can be signed and verified, but also processes that are completed by persons. Public organisations offer processes to citizens for them to complete. It would be highly valuable for

the integrity of the process to be verifiable as well, to be certain that the person, who made these decisions regarding a process in a public organisation really made these decisions and it could verifiable after 50+ years, if a judicial matter arises. When i-voting is used during elections, this new way of signing can be very useful for verifying the integrity of a vote. Though, I am not arguing the lack integrity of Estonia's current i-voting system.

Technological integrity measures are not only for verifying digital data, but also for eliminating trust in organisations by creating evidentiary value. Technology can be used as a deterrent for public and private sector information intensive entities. Situations, where internal secret information is manipulated can be a problem of the past, when modern technological integrity solutions are used.

Digital signatures are not to be confused with digitised signatures- a way of signing for example packages when receiving them, one would physically have to take a pen type of device and sign their name on to a screen. This of course does not have very strong legal power since anyone can sign for this and the person signing can be fooled into thinking that they've signed for a package, when in reality they signed something else into someones name. Digitised signatures can be used maliciously to fool a person. This type of signature does not have any sort of encryption, it is basically a way of signing your name on device, but can be linked to any documents theoretically. Once someone has your signature it can be used for their cause. This can also be used for identity theft just by using the persons signature for their means.

On the other hand a digital signature is a much more secure way of signing documents that is uniquely linked to the signatory; it is capable of identifying the signatory; it is created using means that the signatory can maintain under his sole

control; it is linked to the data to which it relates in such a manner that any subsequent change to the data is detected. These criteria are incorporated into most Estonian systems to allow the person to sign for documents not in the traditional manner of a paper and a pen. This creates an environment, where people who must sign for documents that have legal power can do this from different locations and not physically be there and is a time saver. Digital signatures use public keys that are saved by third parties for incase of future inquiries, that was this person in fact the one who signed the document.

When signing in the traditional manner, one would have to use a device a pen and would physically have to be at the location of signing writing on a piece of paper. One cannot sign from a different location and the person accepting the traditional signature is the one identifying that the person is in fact the person who he/she says they are. The traditional way of signing has legal power as well, so this way of signing has a human element involved. Digital signatures have all the same legal power as a traditional one except that the person does not have to be at certain location when signing for something. They can be on the other side of the world, sign for a document by not physically being there and the signature has the same legal power as a traditional signature.

The reasons for having international regulation on digital signatures is so that we can create a criteria for legally recognising an digital signature by other countries and third parties. Since it creates a way of certifying a document by not being at the location of signing. Regulating this simply means that all the states have a certain set of criteria/rules that must be played by for this system to work and that other states are not creating different means of certifying documents. It also creates fast market access between companies when signing documents that are

legally binding for example NDA or cooperation agreements. Parties don't have to sign documents in the target country.

Auditing of companies and/or organisations can gain a advantage through digital cryptographic signatures. Today, for an information intensive sector to go through an auditing process is strenuous. A whole team of third party auditors visit the company/organisation and must review information to be certain that it has not been tampered with. This of course is a highly ineffective process to gain confidence that electronic data has not been manipulated by a third party or inside defectors. The current process of storing, archiving and verification of integrity is inefficient. Data that is meant to be archived will be done, so in the manner that does not give it very long integrity of information. The auditing process is also taking up too much resources. This process of checking that information has not been changed by a third party is a sign of an un-resourceful government, not knowledgeable of the advanced capabilities around them. Data being stored in physical servers and having personnel waste time on maintaining them should a thing of the past. Cloud computing can enable any party to have low cost IT storage/infrastructure. Cloud has been on the market for a short time, but in most cases, where a reasonable SLA has been made, the service is great. Money is saved, more storage can be bought and changes can be made swiftly.

This main point that can be drawn is that the notion of integrity is misunderstood in the sense that it is though to be a noun, but in fact integrity is an adjective that is acquired through the use of different technological solution and data protocols. This is shown by using different descriptive models, where the conclusion is an adequate model for data integrity, when a data owner stores data in a third party cloud service provider. Perhaps a time stamp cannot give the best integrity

when compared to a digital signature, but no one solution can be the answer, but rather a combination of many for achieving digital integrity in different system.

The work is organised as follows. Chapter 2 talks about the different definitions of Integrity and what is the most suitable for digital integrity. Chapter 3 talks about different scenarios and how digital integrity can be a useful in certain situations. It will give the reader an idea of what the concept of integrity means. Chapter 4 talks about different non-cryptographic and cryptographic technological primitives that can be used by systems to achieve integrity of digital data. Chapter 5 talks about the case the author will be analysing. Chapter 6, which is the bread and butter of the research talks about how an effective solution is achieved, beginning with very simple models and concluding with an effective data protocol for data integrity. Chapter 7 talks about the legal background of digital signatures and where integrity begins, with providing correct information from the start of a process. Chapter 8 will conclude with the result of the analysis.

2 Integrity Definitions

Integrity, when speaking in a non digital context has the following meanings: Integrity from the year 1450, ”wholeness, perfect condition,”from L. integritatem (nom. integritas) ”soundness, wholeness,” from integer ”whole”. Sense of ”uncorrupted virtue” is from the year 1548 [17].

Digital integrity can mean ” what you do when no one is watching; it’s doing the right thing all the time, even when it may work to your disadvantage. Integrity is keeping your word. Integrity is that internal compass and rudder that directs you to where you know you should go when everything around you is pulling you

in a different direction" [18].

These definitions are of course relevant to a person. We would like to examine the definitions of digital integrity. The definitions are as follows:

Integrity can also be defined as "the accuracy and consistency of stored data, indicated by an absence of any alteration in data between two updates of a data record. Data integrity is imposed within a database at its design stage through the use of standard rules and procedures, and is maintained through the use of error checking and validation routines" [19].

Digital integrity as "maintaining the consistency, accuracy, and trustworthiness of data over its entire life cycle. Data must not be changed in transit, and steps must be taken to ensure that data cannot be altered by unauthorised people (for example, in a breach of confidentiality). These measures include file permissions and user access controls. Version control maybe used to prevent erroneous changes or accidental deletion by authorised users becoming a problem. In addition, some means must be in place to detect any changes in data that might occur as a result of non-human-caused events such as an electromagnetic pulse (EMP) or server crash. Some data might include checksums, even cryptographic checksums, for verification of integrity. Backups or redundancies must be available to restore the affected data to its correct state.

Based on the previous definitions digital integrity must have properties that stop an object from being altered by unauthorised persons, have trustworthiness over an extended period of time, must have uncorrupted entities using and manipulating electronic data.

Another crucial factor for data integrity, when speaking in terms of third party storage is proof of possession, in which how can a third party assure that the data

is in their possession and in same condition as it was originally stored.

3 Case Studies

In this part an exploration of potential and past cases will be examined. The aim of this section is to bring to light the multitude of potential risks that can occur in almost every sector, where there is IT dependency and the sector is information intensive. Many condition contribute to the manipulation of data either intentionally or accidentally, but the main risk that is the outcome is the loss of electronic data integrity. For the most part decisions in businesses and organisations are made based on the long period integrity of data.

In the information technology field, there will always be a human factor, where we must trust either insiders in the business or organisation or trust third parties, who are internationally regulatory compliant to accomplish a task put forth. Here we will explore intentional insider and third party manipulation of electronic data.

3.1 Intentional Insider Manipulation Scenario

In more cases than one, for example Herman Simm [20], can the trust of an employee inside an organisation be questioned. Either they have manipulated electronic data or shared data for personal or business gains. It is standard procedure to have an employee sign non-disclosure acts and legally binding contracts that forbid employees from exposing vital business information. For an employee, there are two components stopping them from exposing and manipulating data. The first being a sound contract listing general and specific information that cannot be seen by third parties and an non-disclosure act legally forbidding them exposing

information. Unfortunately we live in a capitalist world and money is the desire of most people in the modern world. The salary of an employee is only as good as the next offer. If we were to imagine a situation where a bribe is offered to an employee for company secrets, if the price is high enough the employee will undoubtedly give the information regardless of the contracts they have signed. But how can we stop this? How can we reduce this risk technologically?

3.2 Intentional Third Party Manipulation Scenario

In this scenario we will examine third party manipulation of data integrity. In the modern information technology field information can be kept in third party storage servers. It is becoming the rising trend to not have a room specially designed for smaller companies with expensive servers to store business information. Instead it is now possible buy this storage at reasonable price and have the third party take care of maintenance. All that is needed for the client is to have a certain level of access to the information. A service level agreement is signed by both parties, but how can the business definitively trust the service provider. Most guarantees are legal and the technological side is left quite unexplored. What guarantees do the company have technologically that their data is not being manipulated by this third party. Here money is always a key factor as in the previous scenario[22].

3.3 Medical Records

In the medical sector as mentioned before the efforts of the European Union for implementing an effective system for transmitting and viewing different patient records from different European Union countries is an ideal area, where digital

integrity should be also implemented. If we have large quantities of digital objects moving from different countries to different information systems, it will be very useful to use the techniques mentioned before to reduce errors either intentional or non-intentional to reduce the amount of medical records errors and ensure the authenticity of information, when being viewed by a third party [23].

A very simple hypothetical example of the need for digital integrity in the medical records sector would be one that can be defined as very personal. Lets imagine that you are someone close to you is in a car accident and required a blood transfusion to survive the accident. Lets imagine that this person is another country. All medical procedures aside and for the sake of argument, the medical records have been altered accidentally unknowingly to the patient. When the patient arrives at the hospital it can be possible that the person receives blood that is not the same as them. This type of situation can cause unprecedented harm to the patient and the people close to them.

This type of situation can easily resolved by using either Message Authentication Codes to ensure that the electronic data incoming from the sender is authentic by checking the output or message digest. It is useful because private medical records should not be viewed and it would make it possible to have the person responsible to only check the digest instead of viewing the entire medical records and comparing it to an original. It would also be useful to use Error Detection Codes to make sure that not any information was lost during the transmission of the message of the file.

A possible solution does not only have implication on data integrity but more over on the lives of the citizens themselves. Currently at least in Estonia a citizen would show their state issued ID-card and the information will be queried from a

state information system. But if a person is abroad, then the medical information relevant to them is not as available to them as in Estonia.

3.4 Server Storage

Server storage is another problem that could be solved by using digital integrity solution methods. When comparing the past and future in terms of information storage, the past had a lot more elbow grease involved in the storing and checking authenticity of records or information. In the past we were more concerned with paper document management. The space used was enormous and the time to look a document up would have taken ages. We had to trust the authority doing this type of job and hand checking authenticity of information. In the modern world the amount of data collected and stored is increasing at an exponential rate. The use of servers either in physical form on-site or using a third party to store data is the way to go. Usually smaller businesses would use a dark room to store a few computers (servers) to store their data. This, depending on the sensitivity of the information is unfortunately also becoming a past practice. With third party cloud providers able to give large amounts of server storage, it is becoming redundant to store data on site[22].

In this scenario we would use a third party cloud storage service provider to store the needed data in a place unknown to use. But does it really matter where it is, it only matters that after the point in time that the data is stored, that the data itself will keep the integrity and authenticity after the time of storage. In this scenario, it will be useful to use cryptographic hash functions to encrypt the data and use the message digest to check, if the data has been altered by third parties[7]. It is of course in the interest of the third party storage service providers

to ensure that data is not manipulated in their information system, but this will not stop third party intentional manipulation of data. It can be used as a deterrent moreover to ensure that data stays authentic during longer periods of time[22].

3.5 Telecommunication

Currently all data in the European Union is being retained for a period of 6-24 months that is transmitted through an Internet Service Provider (ISP), which is required by the European Union Data Retention Act. This information is used for analytical purposes in the event of criminal activities. This retention of information is done without the consent of European Union citizens, but that is not the topic of this paper. With information running through ISP's and also depending on the target and sensitivity of the information, it is possible to alter the data in while it is transmitted, which opens up a world of data manipulation possibilities. It is useful to use Error Detection Codes to deter possible third party manipulation of this happening[24].

3.6 Updating

The automobile industry is seeing some serious developments, as it is also a heavily IT dependent. It is only a matter of time before we see hackers try to manipulate cars while on the road. There is very interesting News article in the International Business Times, where a journalist finds out first-hand what is possible, when IT dependent road cars information integrity is compromised. This research was interestingly funded by DARPA (Defense Advanced Research Projects Agency)[25] .

In the article it was shown how vulnerable a car is by manipulating the steering, displays that show speed, warnings, fuel capacity and many other parameters. Although this was a controlled environment and everyone was aware of what was happening, one cannot comprehend what could be possible and the fear a person is put through, when this happens unknowingly to them. Tesla is also developing electric cars that are heavily IT dependent and can be updated using the 3G connection in the persons smart device. The update file is susceptible to data in transmit manipulation. This is a scenario, where message authentication and error detection code can come in handy, when verifying the update file or any information that is sent from the car manufacturer.

What can be concluded from these examples is that integrity can have many definitions and uses in different situations. Due to this, in this work the author will concentrate on one specific example, which is the the scenario of what integrity is when speaking of a data owner, who has trusted a third party cloud provider to store their electronic data. The author will analyse the different situations and disputes.

In todays world, where technology is making leaps and bounds and has effectively created an environment, where no place is out of reach and close to, if not all, sectors have become information technology dependent. If we were to think of a certain business sector, where information technology is not playing a key role- it will take a while to figure out an appropriate result. This situation, where civilisation has evolved from paper document management to digital document management has been creating the groundwork for a new technologies to be utilised to contain the integrity of exponentially increasing amounts of digital data that is being created every second. I recall from my 4th semester lecture with Mr.

Raul Rikk, where we were shown a picture of the National Library in the United States of America. At one point in time before the age of digitalisation, that library held almost all of human knowledge with those walls. It was only when I researched myself to get an idea of the magnitude of the building and it was immense. For our intellectual pleasure, Mr. Rikk gave an idea of the amount of time it takes for the NSA to collect the same amount of data- just six hours. As mentioned before, the groundwork for new technologies is being lain and the amount of electronic data that could potentially be secured in the context of integrity is immense.

The application of such a solution can have a large impact on the information communication technology intensive sector in many fields. In the products and service field, there are three main useful aspects of a potential integrity solution. The elimination of insider trust creates an environment, where the cyber actions of employees inside a respective company can be verified. For example, a person manipulated information, be-it intentionally or on accident- this is a case of digital forensics after the action has been completed. There exists certain third party organizations, who can verify the integrity of of electronic data during an audit. A possible solution can reduce the amount of time and finances required to perform tedious audits. As I mentioned before many sectors are becoming IT dependent and it is only a matter of time before it spreads to other fields.

Companies and organisations acting in the international market must be compliant with laws/processes that are put forth by monitoring organisations. A possible solution can verify the integrity of a digital process, so third party auditors needn't come to the respective company/organisation to physically perform auditing tasks, but a simple log file with verified process signatures can be sent to

the auditing company for verification. This can create financial benefits on a large scale. Then again, it makes personnel redundant in the face of IT solutions and employees can loose their jobs.

Generally in the ICT sector a potential technology can create value with independent authentication of digital objects. This creates a mindset that any digital movement can be verified potential manipulator is deterred from his actions. In the future data could be kept safe and information can be trustworthy.

4 Technical Primitives

This section describes different technical primitives, presenting both non-cryptographic codes for information integrity. Non-cryptographic codes are described using error detection and error correction codes. Cryptographic codes are described using hash functions message authentication codes and digital signatures.

4.1 Non-Cryptographic Codes

In todays information technology systems with important information surging through, the checking of incoming and outgoing information is crucial even through unreliable communication channels. Some information can be corrected using ECC memory or Error Correcting Code memory due to channel noise and due to that fact errors may occur during the transmission. Error correction can allow for the message to be reconstructed similarly to the original message and error detection allows for the error to be detected. ECC memory is able to correct single-bit errors with the user unaware of the process, but ECC is unable to correct double bit errors an under the assumption that the information in the system is important

then the system would need to stop working with incorrect data. The change in information according Jeff Layton occurs with some sort of electrical or magnetic interference which causes the change even seemingly randomly [26].

4.1.1 Error Detection

Error detection is commonly used by utilising a hash function. Error detection can be used by adding a hash function to the message and computing the data into hash function upon receiving and comparing the hash function to the on sent to authenticate the message. Repetition codes are used by repeating bits through a communication channel. If we were to send a 1001 to a recipient, then the code is sent by three blocks 1001 1001 1001, now if the receiver sees 1010 1001 1001, then it is evident that an error occurred. Parity bits are used by adding an extra bit to ensure that the outcome is either odd or even. Checksums are modular arithmetic sum of message code words of a fixed length. Cyclic Redundancy Check is an insecure hash function designed to detect accidental mistakes and changes to digital data.

4.1.2 Error Correction

Error Correcting (FEC) allows for controlling errors in transmitting data using an unreliable channel or communication channels that are noisy. The idea behind Forward Error Correcting is that the sender would use and encoding method that is redundant by using Error Correcting Code. This redundancy allows the receiver of the message to make out a limited number of errors that may or may not occur in the message and in many cases correct the errors without having to retransmit the message. This technique gives the sender the possibility to not retransmit the

message, but the only downside is that the forward transmission must be a higher bandwidth. This technique is normally used on mass storage devices, where it would be necessary to recover corrupted data an it is also used in modems.

A study shows that of real memory check took place at google and during the study they found that a third of their machines and 8 percent of Dual in-line Memory Modules witnessed correctable errors on scale that was though to be much lower than before. Google is experiencing about 25,00075,000 correctable errors (CE) per billion device hours per megabit, which translates to 2,0006,000 CE/GB-yr (or about 250750 CE/Gb-yr)[27].

4.2 Cryptographic Codes

4.2.1 Hash Functions

Cryptographic hash functions are used by taking an arbitrary sized digital data and return a fixed message hash, which has the following properties:

- *Non-invertible*– given a hash, it is infeasible to reconstruct the data.
- *Second pre-image resistance*– given data, it is infeasible to find different data with the same hash.
- *Collision freeness*– it is infeasible to find two different data items with the same hash.

The first hash functions were MD2, MD4, MD5 and SHA. The latter was proposed by NSA in the early 90's. They are followed by SHA-1, SHA-2 and SHA-3 algorithms that correct for weaknesses in the earlier algorithms [28].

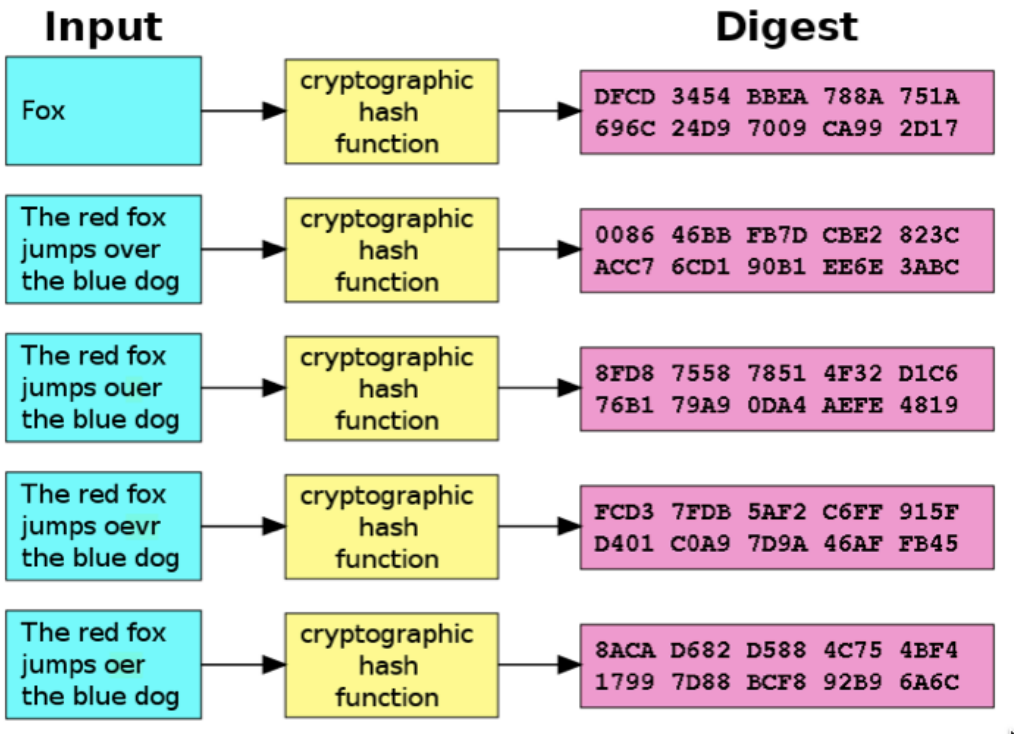

Figure 1: Hash digest calculation process.

In Figure 1 describes how Cryptographic Hash Functions notably SHA-1 works. The first box is the input message, it is hashed using SHA-1 and a the message result or message digest can be seen in pink. Even with the message input changing by one letter the result of the digest is very different. Giving this technique a very useful application to message integrity. It is also close to impossible or very tedious to change the message without it having an effect on the digest. This technique is mostly used by digital signatures and message authentication codes which will be talked about in the next section. Cryptographic Hash Functions have application in the digital integrity sector in which a sender or receiver can

validate the integrity of a message by calculating the hash value and comparing the calculated hash with a securely stored copy.

This technique also has application purposes when storing passwords. It was first invented by Roger Needham, who was a British computer scientist. He proposed that instead of storing passwords in a text files, using cryptographic hash functions, only the hash digest would be stored and the password hash would be checked against the digest. If a password was forgotten then it would need to be replaced with a new one. It is an effective way of storing information in the sense that it is near impossible to find out the password if trying to decrypt the hash digest, since hash functions are one way. It is an effective password verification technique [8].

4.2.2 Message Authentication Codes

Message Authentication Code or MAC is a minuscule piece of information used to authenticate a message assurance of integrity and authenticity. MAC is able to detect changes in a message and can gives assurances of the origin of the message. A MAC algorithm can also be called a keyed cryptographic hash function, it accepts a secret key as an input and a random piece of the message to be authenticated. The message authentication code allows the verifier to control the authenticity and origin of the original message. This technique does not allow a non-repudiation feature of the message due to the fact that both parties have a private key. By non-repudiation, we mean receivers ability to use the received data as evidence against the sender.

4.2.3 Digital Signatures

A digital signature is a cryptogrphic model for demonstrating the authenticity and of a digital message or document. It gives assurance of the origin and the senders is unable to deny that the data from them(non-repudiation).

In Figure 2, the processes of the creation and verification of a digital signature are depicted. It is created by having the message or electronic data being hashed and the result is the hash digest. The message digest is then encrypted using the sender's private private key. A certificate that is issued from the Certification Authority is also encrypted with the message and this is then a digitally signed document/data. For the verification process the data hash and the signer's public key hash are compared, if the hashed are equal, then the signature is valid, giving authenticity to the reader that this message came from a specific person and the message has not been altered after the signing.

Lets imagine that John would like to send a message to Jane and digitally sign it. First what John must do is create the message or document he would like to send to Jane. After he is finished, he needs to sign with his ID-card PIN codes or his Mobile-ID PIN codes either way he can create a digital signature. When a message is created it is pushed through a hash algorithm, which then creates a hash value for the document. John can then encrypt the message using his private key. The certificate he received is checked to verify that it is active and this is hashed as well. All this is then attached to the document and a signed encrypted document is created. During the creation of the signature, it is also time-stamped. Essentially Public Key Infrastructure is a mechanism for verifying that the senders message came from the sender themselves and not from any other third party.

Now For jane to verify that John actually sent this document to him and it has

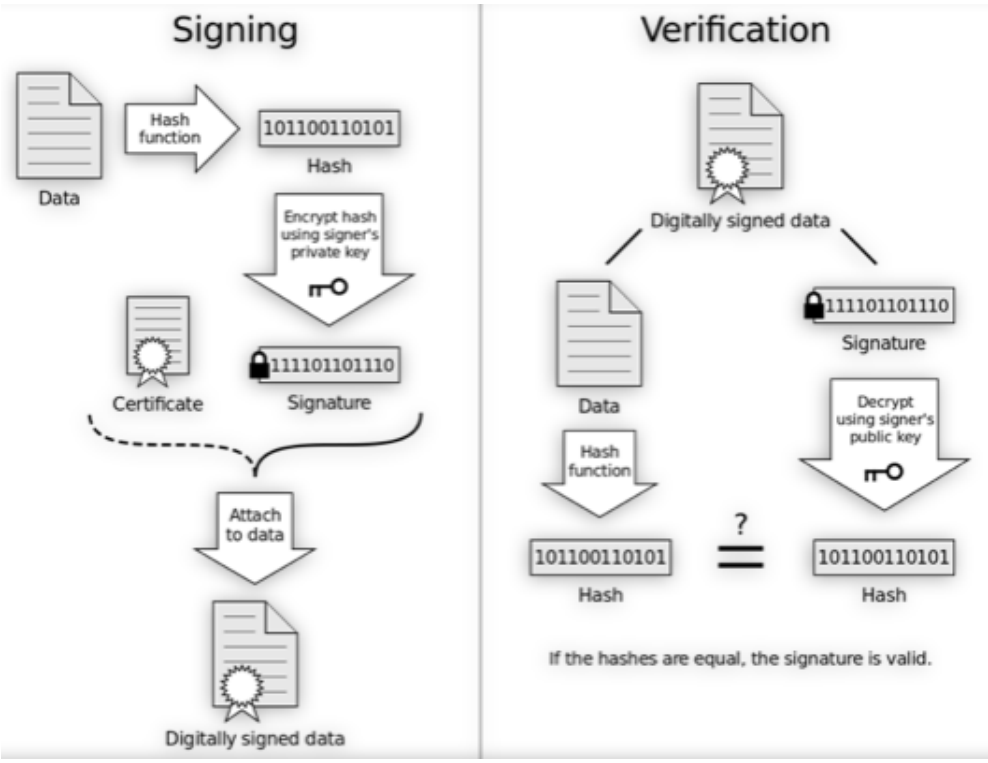

Figure 2: Digital signature creation (left) and verification (right).

not been changed, she will need John's public key. If the hash functions from the data and signature are the same then it can be confirmed that the signatures are valid.

An example can also be that John loans Jane 1000,00 EUR. Alice signs a contract that she really did receive that amount in the interest of John, so if a situation, where it must be proved in court occurs John has no problem proving the transaction. But what happens when Jane's certificate is long revoked and requires authenticity that indeed she did sign with an authentic and active certificate? Then it is evident that electronic documents should be able to prove their authenticity

long after their creation.

4.2.4 Time Stamps

There are two types of time-stamping services: Figure 3 is the process of a signed time/stamp being created. The second Figure 4 Time-stamping can ascertain whether an electronic document was signed and created at a certain point in time, without time-stamping today and the amount of electronic data being transmitted everyday, we cannot trust a system in which the signer can repudiate a signed document themselves [1].

Signed Time Stamps Figure 3 is a simplified illustration of how a time stamp is created and the parties involved. If a party sends a hash corresponding to data, the TSA (Time Stamp Authority) sends a signature with their time stamp. This proves this data existed at this point in time.

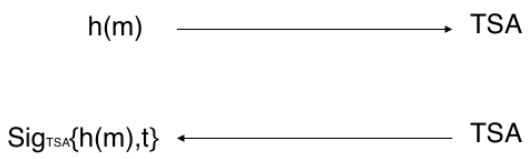

Figure 3: Signed time-generation procedure.

Linked Time stamps were first proposed by Haber and Stornetta [29]. Linked time-stamping is way of creating tokens that are dependent of each other. Altering data later causes the entire structure to become invalid and altering of data can be

detected. Haber and Stornetta later improved the efficiency of their schema by using hash trees 4 to create per-second global hash values for all the documents time-stamped over the globe [1].

The so called linked accountable time-stamping [4, 8] uses hash calendars to establish more tight relationship per-second hash value and physical time. The goal was to reduce the trust in a third party trusted entity and so digital object could be independently verifiable. Figure 4 is an example of a hash tree and calendar. They are methods of having one way signing of digital objects and to have a root hash value for checking the integrity of a hash value in the past. Every dot represents an event of singing a document, if the root hash does not give the value of the singing event then manipulation of data has occurred. In [8] time-stamping is defined as a set of principals with the Time-Stamping Authority (TSA) and Publication Authority(PA) with four other protocols (S, C, V, P). The stamping protocol S is used is used by a person to hand over the message to the TSA for time-stamping. During the stamp completion protocol C the person obtains a time certificate from the TSA. The verification protocol V is used to verify the temporal order of the stamps using two time certificates. The publication protocol P is used by the TSA to handle a round-stamp to the PA who will publish who will publish it on some authenticated and easily accessible medium [8].

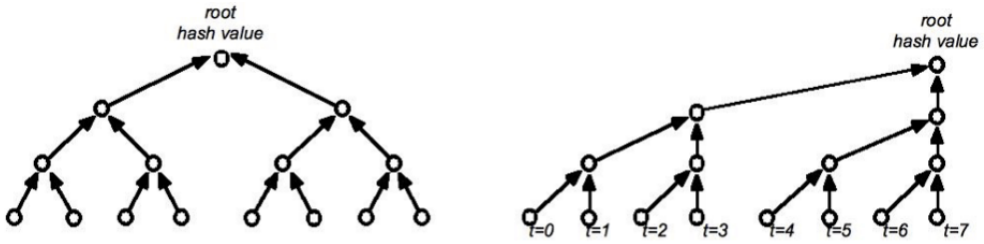

Figure 4: Hash tree (left) and hash calendar (right). Also called linked time-stamping.

Time-stamping can also be defined as a service of protocols providing long-term authentication of digital documents together with the moment in time at which they were submitted for authentication [8].

An effective time-stamping system can be deemed accountable, if any of the parties in the next situations, one is uncorrupted:

- *Fraud detection*: The service makes the trusted third parties accountable for their actions by enabling a principle to detect and later prove to a judge any frauds affecting the relative ordering between time-stamps [8].

- *Anti-framing*: If a party has honestly followed the protocol, but is still accused of forgeries, they can explicitly prove any false accusations [8].

- *Reordering*: Time-Stamping Authority assigns an earlier stamp to a document that was submitted later than another document. The TSA should provide Relative Temporal Authentication [8].

An effective time-stamping mechanism can assist in making the current Public Key infrastructure trust level lower by proving that an electronic document was

signed before the revocation of a signature key. The accountability must run through the Time-stamping Authority as well- making the authority responsible for any actions that require verification over time [8, 9].

5 Analysis of Cloud Case Data Ownership

In this section we will analyse a specific case related to data integrity using a cloud service provider and a data owner, who stored their data with a third party. The scenario is as follows: a client would like to store their electronic data using a cloud provider as a trusted service provider. The service user will want to if that integrity of the electronic data is kept and if any disputes arise then how will the technological primitives be used to solve these issues. Normally, if an organisation or entity would like to use a third party cloud service provider, then an adequate Service Level Agreement (SLA) is signed by both parties ensuring that the level of availability, confidentiality and integrity is met. An SLA can also be an ineffective mechanism for solving disputes, which is why an effective technological mechanism must be used in order to determine who is responsible for electronic data errors.

As mentioned before in this case there are two parties which will be analysed, the first is the user and the second is the cloud service provider. The user or data owner is an entity with electronic data, who wishes to store the data in a third party cloud service providers servers. The users interest is to not store all of their information on-site and that the cloud provider will store their electronic data with integrity intact, which means they will not be changed by unauthorised users for the agreed upon time, unless the user sees fit to change the information

themselves. The user may also want to check the authenticity of the data they stored, if it is still as they agreed upon and if they are in fact still in the cloud service providers servers. Most importantly the user wishes that the cloud service providers is responsible for their actions in regard to their data.

For the user the cloud service works on an application, where it asks for information and this information is then changed according to how the user has used it and then is finally saved in the cloud service providers servers again. The service for the user is not free and must compensated financially for the cloud provider depending on the services rendered. In both of the parties interest, they wish to provide good service, but for the cloud provider the storage of data is an expense and we can imagine that the cloud provider may take short cuts to save money by for example erasing parts of data. If the cloud provider does not keep extra copies or follow predetermined processes then information can be lost.

Lets assume supposedly that a dispute arrises in the event that the user claims that a piece of data has gone missing. The user can claim that the cloud service provider has not stored data adequately based on the users own control of the data in the cloud. In the event of this claim, it of course can be true, but on the contrary can also be malicious acts on the user side. Disputes like this must be solved and a convenient solution for having proof could be the answer. This study will not concentrate on legal fraud that can be carried out by the user, but it will concentrate on an honest technological solution, so incase of such a claim, it must be capable of proving its authenticity. On the other hand if the cloud provider claims that in fact the data integrity of what was stored has not been changed then they too must be able to provide proof.

In the following models we see that a party has the obligation to prove that

the actions they took were authentic. This is called the evidentiary burden. In the following examples regarding the data owner and the cloud provider in the different models the evidentiary burden depending on the model examined will fall in wither the data owners or the cloud service providers obligation.

6 Solutions Using Technical Primitives

This section describes how solutions using codes, non-cryptographic codes and cryptographic codes can give integrity digital data. Solutions will be examined by three protocols: (1) normal usage, (2) verifying, and (3) if a dispute can be solved by both parties having the possibility to provide proof of possession in a court of law.

6.1 Solution Using Codes

A solution utilising non-cryptographic codes seen in Figure 5 is also possible using error correction and detection codes. This solution is the simplest for achieving data integrity in the context of third party cloud service providers. Along with the data being uploaded into the cloud, a small portion of code uploaded along with it, being it either for error detection or correction.

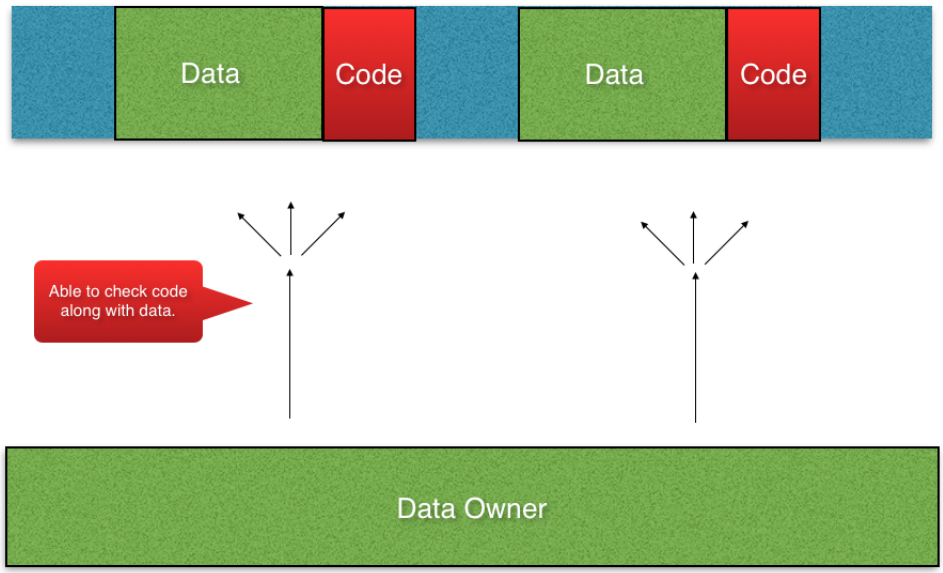

Figure 5: Solution using codes.

6.1.1 Normal Usage

Normal usage of this solution can detect situations, where there are accidental mistakes in stored data.

6.1.2 Verification

This solution does not help to detect changes made by an adversary.

6.1.3 Dispute Resolution

This solution cannot help in legal disputes against parties because no verification is provided be either party.

6.2 Solution Using Hashing

When storing data in a cloud service providers servers, it is also effective to continuously check the integrity of the data itself. Figure 6 is a development of the previous in the sense of checking original data's integrity. Rather than having an entire data represented by on block, it is also possible to incrementally hash the data into smaller blocks for more effective integrity checks. If the data owner wishes to check integrity for example of $D3$, then a calculated hash value is sent to the cloud service provider to provide authenticity of the data.

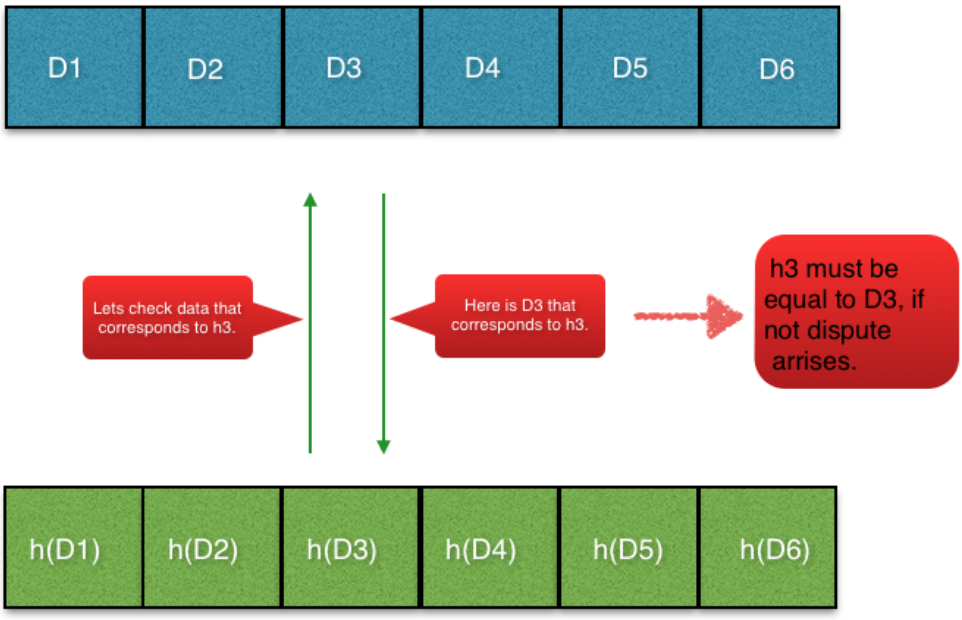

Figure 6: Verifying data with incremental hashing.

The cloud service provider sends the corresponding data to the data owner for

them to authenticate if,

$$D_3 = h(D_3) \ ,$$

if the data provided by the cloud service provider calculates to an equal hash of the data owner then integrity has been kept.

$$D_1,\ D_2,\ D_4,\ D_5,\ D_6$$

can also be have integrity checked using the same process.

6.2.1 Normal Usage

Normal usage of this solution can detect situations, where there are mistakes in stored data.

6.2.2 Verification

This solution can be used to verify data integrity by the user due to that fact that they can keep the hashes as evidence.

6.2.3 Dispute Resolution

This solution cannot help in legal disputes against parties because the cloud service provider is able to alter data and produce a new hash, without client authentication signatures, it is close to impossible to tell if data was altered by an authorised user or the cloud service provider.

6.3 Solution Using Server Signatures

We incorporate into Figure 7 server signature. Using this the data owner can provide evidence if a dispute arises. The Server signature are sent by the cloud

service provider as evidence that the data sent by the user was received.

6.3.1 Normal Usage

Figure 7 represents the first model of the technical solution for data integrity. Here we can see how the process works. First the Data Owner wishes to store data on the cloud service providers servers. The Data Owner uploads information and also uses hash functions to hash the data and in-turn produce a hash value. The cloud service provider digitally signs the data and resends their hash with the signature proving that they have received and stored the data owners information. A signed hash is sent back to the Data Owner.

6.3.2 Verification

The data owner can verify which where the data owner can compare the hash that was produced by them and the cloud service provider. If the hash match then the integrity of the data has been kept authentic, when the hash created by the Data owner and the data sent from the Cloud Service Provider hash are the same:

$$h(D) = h' \ .$$

The data owner stores the signed hash on site, which are called cryptographic attributes.

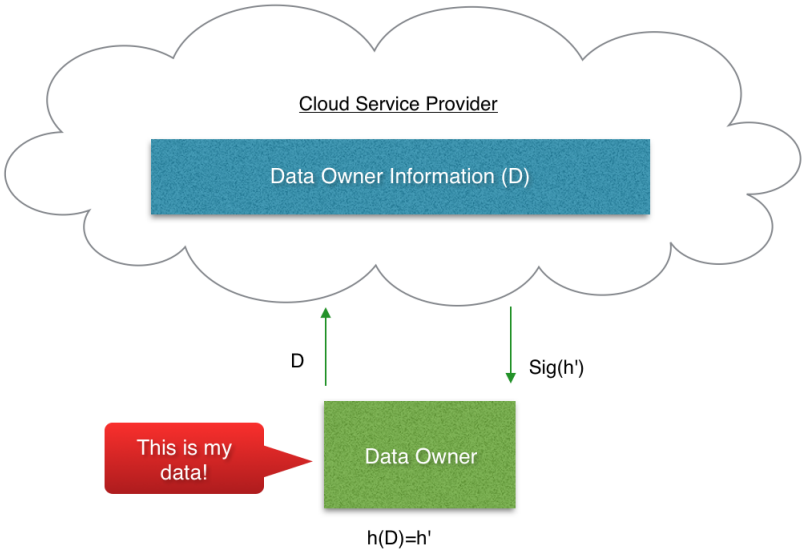

Figure 7: Integrity protection using server signatures.

6.3.3 Dispute Resolution

In figure 8, we look at when the data owner wishes to see the information that has been stored by the cloud service provider. We will also look at how a dispute is settled, if the hash sent by the data owner is not equal to the hash calculated from the data that the cloud service provider has sent. This happens rarely, but in context of analysis in is necessary to view every angle to achieve an adequate schema. The data owner requests to view information corresponding to a hash sent by the cloud service provider.

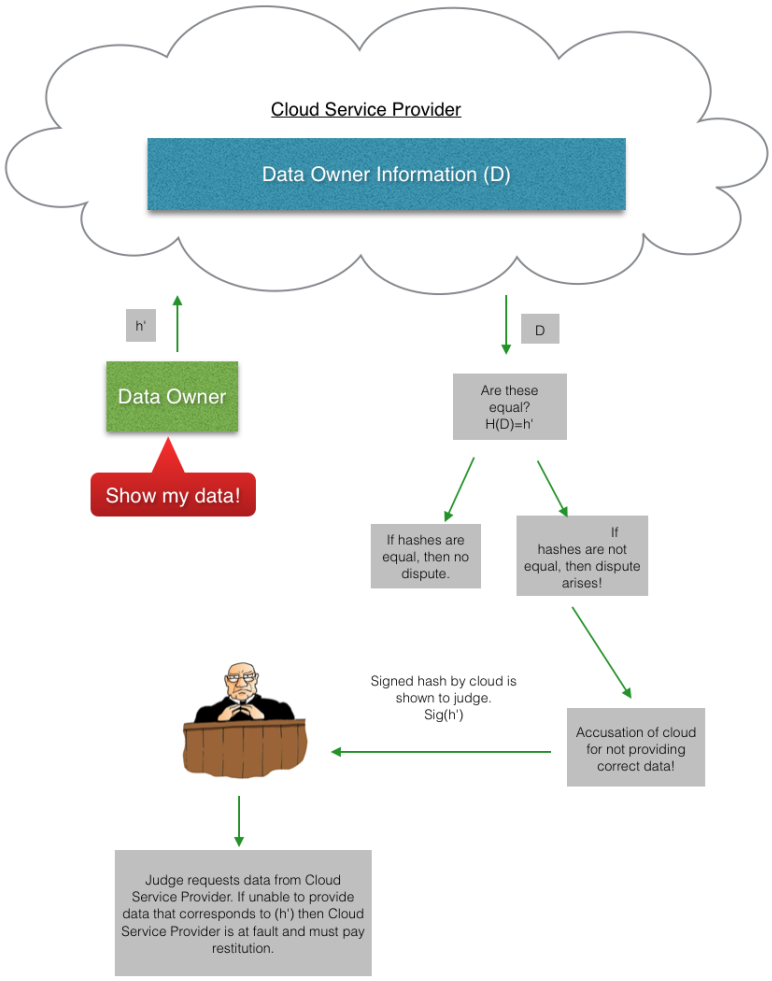

Figure 8: Dispute resolution using server signatures.

In this case if the hash calculated by the data owner of the sent information is equal- meaning the data sent by the cloud service provider is also calculated into a hash, then the two must be equal. In the event that these are not equal, the data owner can resort to taking the matter to court. Accusing the cloud service

provider of not supplying information corresponding to the hash calculated by the data owner. In this model the evidentiary burden lies with the cloud service provider. If the cloud service provider is unable to provide data corresponding to the hash that a judge is requesting then the cloud service provider is at fault and must pay restitution to the data owner for not being able to supply correct information corresponding to the hash sent by the data owner. Also in this model it is important to note that the only party signing information is the cloud service provider, this leaves us with an unfair model for the benefit of the data owner.

6.4 Solution with Multiple Changes

6.4.1 Normal usage

In Figure 8 the data owner was only storing data and requesting to see it in the future. But using a cloud service provider can enable one to also store and change the stored data. In Figure 9, we see that the data owner has recently stored data and wishes to make a change to the data. In this case the data has selected a portion of the data and has changed it.

The hash calculated from D can no longer be equal to the hash calculated originally, since it has been changed by the data owner. In this case the data owner selected a portion of data and now a new portion of data is saved with the original D representing a change to the original data. It is represented by D', further the cloud service provider sends a signed h' back to the data owner representing that the change made has been documented. The data owner is now able to check the authenticity of D' by requesting to see the data by sending h'. the cloud provider is obligated to send data that corresponds to the D'. Also note that the cloud

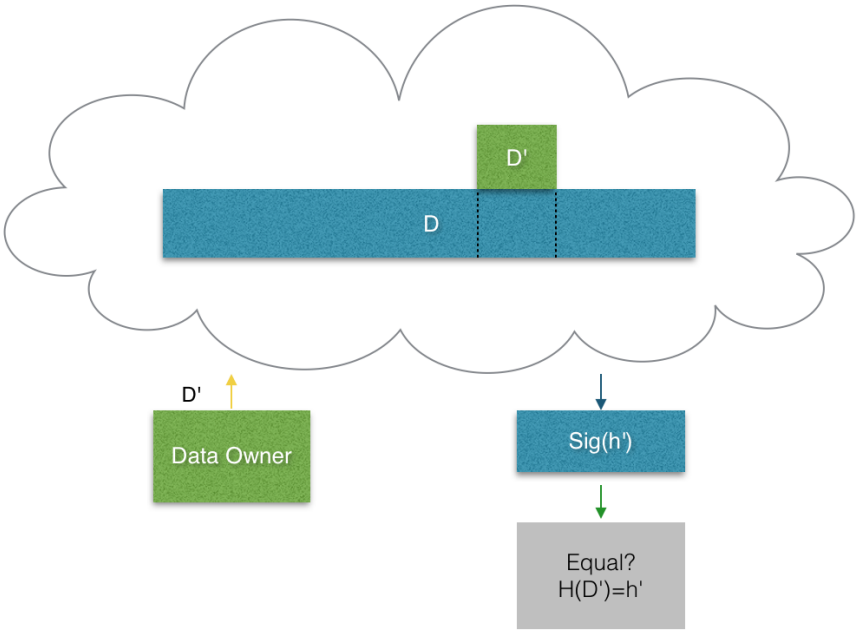

Figure 9: Solution for checking changes to stored data.

provider is currently still the only party sending a signed hash back to the data owner, which the data owner can store as evidence. The evidentiary burden still lies with the cloud service provider and this model is still not balanced equally for bot parties.

In Figure 9 the data owner was able to make a change to the original data and received a signed hash back from the cloud service provider. In Figure 10 the data owner has made a sequent change to the data that was changed before hand.

The most recent change to data is represented by D''. Now the data owner has 3 signatures from the cloud service providers as evidence of stored data

$$\mathrm{Sig}\{h\},\ \mathrm{Sig}\{h'\}\ \mathrm{Sig}\{h''\}\ .$$

The data owner can request data based on these 3 signatures. The cloud service provider is obligated send the data owners information that correspond to the three signed hashes provided by the cloud service provider to the data owner. The data owner can check the integrity of the changes made by comparing the 2 hash values. In Figure 11 the data owner has made 6 changes to the data that was

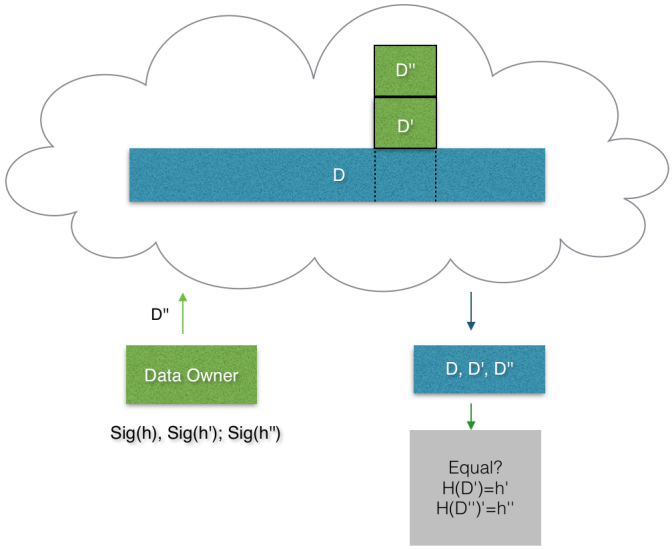

Figure 10: Multiple changes to stored data.

originally stored represented by

$$D'_1,\ D'_2,\ D'_3,\ D'_4,\ D'_5,\ D_6\ .$$

The cloud service provider has sent signed hash values to the data owner showing that they have documented the changes cryptographically with

$$h(D),\ h(D'_1),\ h(D'_2),\ h(D'_3),\ h(D'_4),\ h(D'_5)\ h(D'_6)\ .$$

It is important not that the only party authenticating changes to data is the cloud service provider. All the changes or cryptographic attributes are stored by the data owner for future authentication of data integrity. It is possible to save all changes made by the data owner and in the next model we will examine how this is not an effective model.

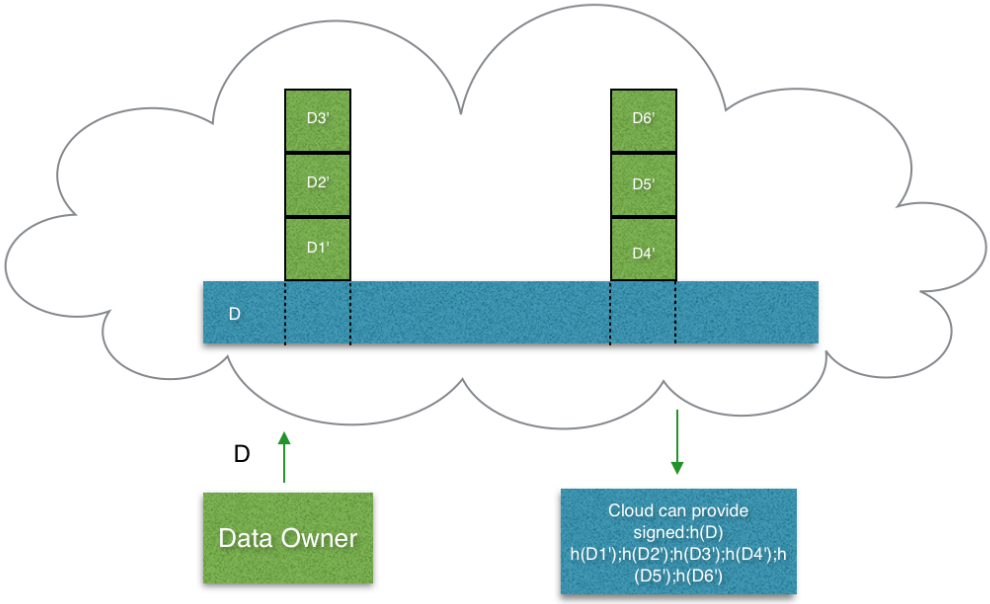

Figure 11: All changes to data have been saved and all corresponding signatures, which also include signatures of past changes are kept by the data owner as evidence.

6.4.2 Verification

Figure 12 represents a development from Figure 11, where the data owner has made subsequent changes to data and has received all signed hash values repre-

senting the changes made to the data. In the interest of storage effectiveness the cloud provider has decided to erase past changes and save only the latest data believing that the latest data is what is required for the data owner. The cloud has provided signed hash values for

$$h(D),\ h(D'_3),\ h(D'_6)\ .$$

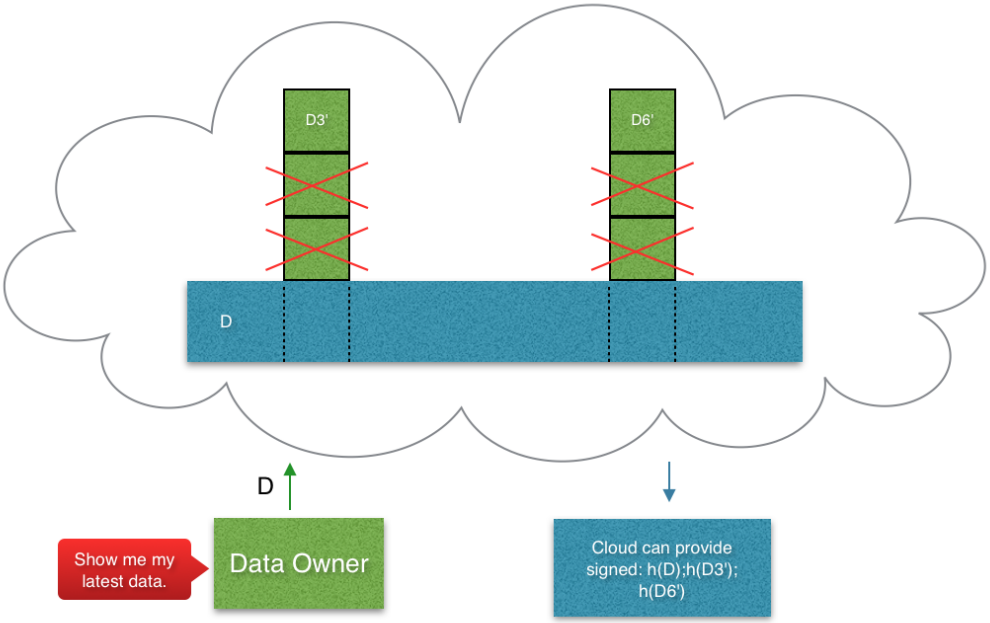

Figure 12: Data owner wishes to verify stored data, however the cloud service provider has deleted past changes for more effective storage.

6.4.3 Dispute Resolution

Figure 13 represents what can happen, if the cloud service provider in the context of storage effectiveness has decided to erase past data changes in the belief that

the latest data is what is required by the data owner. If the data owner even being unaware of the deletion of past data requests this data based on the signed hash values from the cloud provider can cause complications in a dispute for the cloud service provider. The data owner now armed with all signatures from the cloud service provider requests data corresponding to signatures received from the cloud service provider. Since the past data has been erased the cloud service provider cannot provide the data requested by the data owner. The matter can again be taken into a court of law and the data requested from the cloud service provider by a judge. If the data corresponding to the signed hash values cannot be provided by the cloud service provider then compensation must be paid to the data owner for not providing correct data. This model is ineffective for the data owner for not being able to use past data and for the cloud service provider for being continuously being the only party providing evidence of data changes.

Schema 1 Figure 14 is a continuation of the previous model explaining the lack of a dispute schema in court matters of data changes. If a dispute is taking place in court then how the process plays out is important for both parties. In this case the data owner has provided to a judge:

$$\mathrm{Sig}\{h(D''')\},\ \mathrm{Sig}\{h(D'')\},\ \mathrm{Sig}\{h(D')\}\ .$$

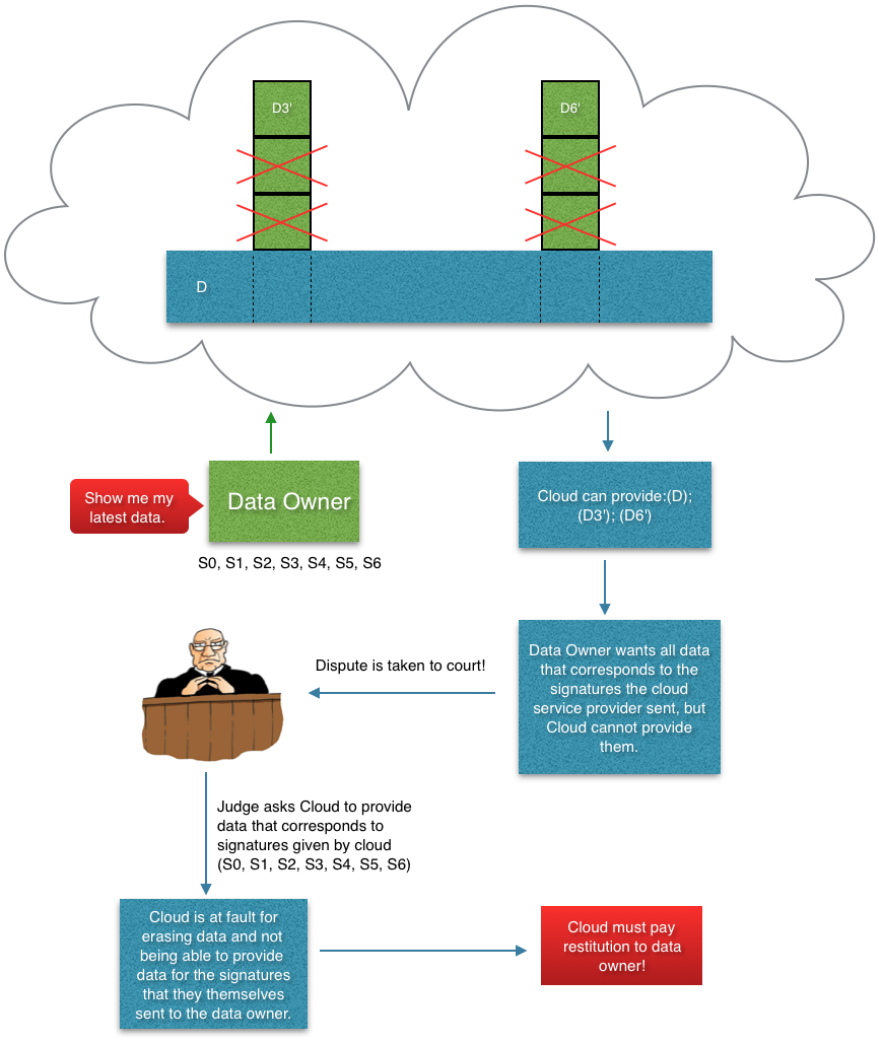

Figure 13: Dispute resolution, when the cloud service provider decides to erase past changes in the mindset that the past data is no longer relevant to the data owner.

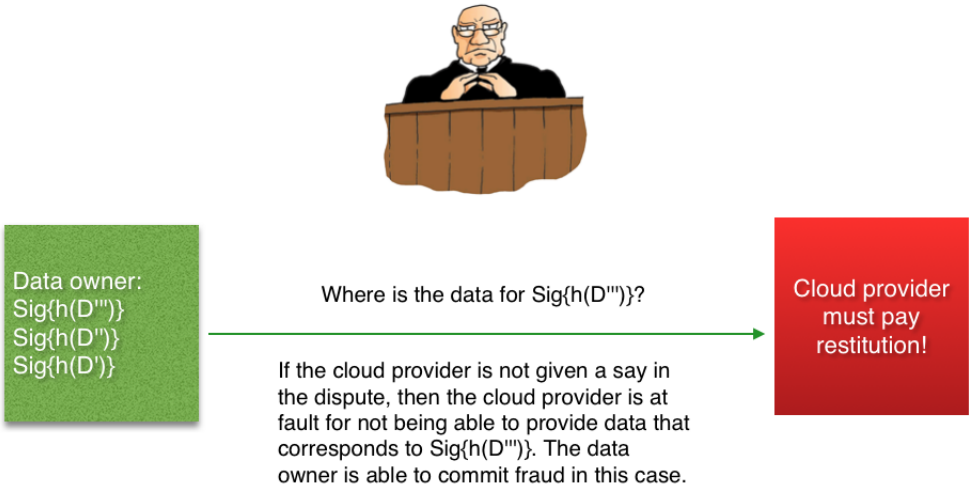

Figure 14: Dispute resolution situation, where cloud service provider is not given opportunity to defend and must pay restitution to the data owner.

The judge can view the signatures as definitive evidence and give an opportunity to the cloud provider to defend themselves. The judge will rule in favor of the data owner and the cloud service provider must pay compensation for not being able to provide data that corresponds to the signatures shown to the judge.

Schema 2 In Figure 15 the data owner has provided the judge with a set of 2 signatures $\mathrm{Sig}\{h(D''')\}$ and $\mathrm{Sig}\{h(D'')\}$. The judge requests the data corresponding to the latest signatures provided by the data owner. Now when the judge requests data corresponding to the 2 signatures, the cloud service provider is given an opportunity to have a say in the matter. This time the cloud service provider is clever and claims that the 2 signatures provided by the data owner are in fact not the

latest data changes that they have received. The cloud service provider can fabricate a third signature, which proves that the data owner was attempting to commit fraud.

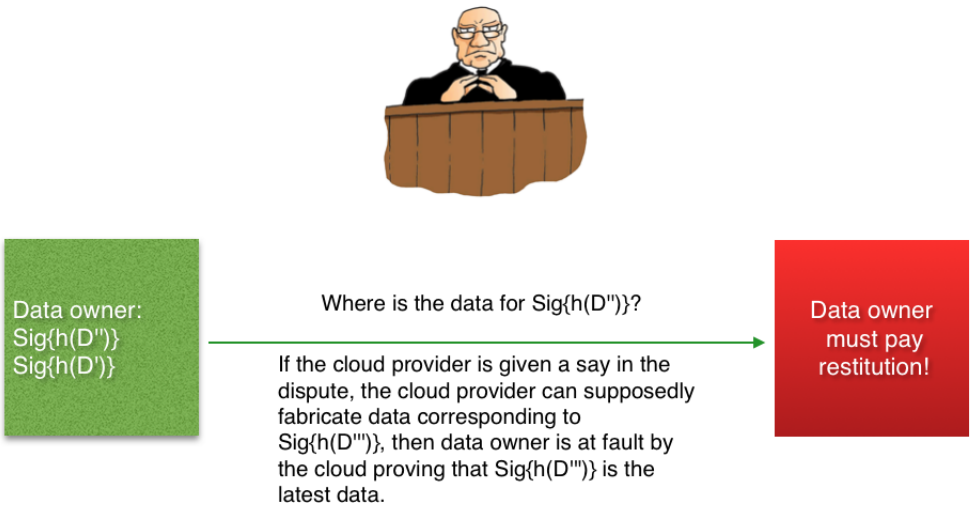

Figure 15: Dispute resolution situation, where cloud service provider is given opportunity to defend and can potentially fabricate a latest signature to provide evidence of data owner committing fraud.

6.4.4 Conclusion

Figure 14 and Figure 15 prove that it is impossible to differ from data that was fabricated by the cloud or if data was indeed sent by the data owner and it corresponds to the signatures provided by the cloud service provider. All past illustration of models proves that client authentication is required. This can enable the cloud service provider to store evidence in the form of signatures in the event of a dispute or

accusation from the data owner. The cloud service provider can provide evidence and the burden of proof now can lie with both parties as seen in Figure 16.

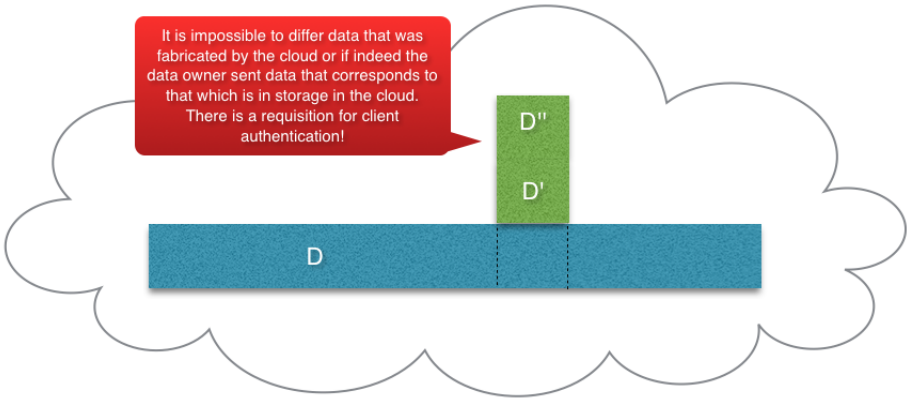

Figure 16: It is impossible to differ from data that was stored and signatures that were created by the cloud service provider.

6.5 Solution Using Client Signatures

6.5.1 Normal Usage

Figure 17 has incorporated authentication from the data owner side. In this case, when the data owner wishes to store data, then the data must be accompanied by a digital signature from the data owner. As in the illustration below, the data owner is changing the originally stored data D. The change is represented by

$$D' \ ,$$

but now a signature allowing the authentication of the change is also sent by the data owner. the signature is stored along with

$$D', \mathrm{Sig}_U\{h(D')\} \quad .$$

This signature is kept as evidence by the cloud service provider in case of a dispute and if they are left with evidentiary burden.

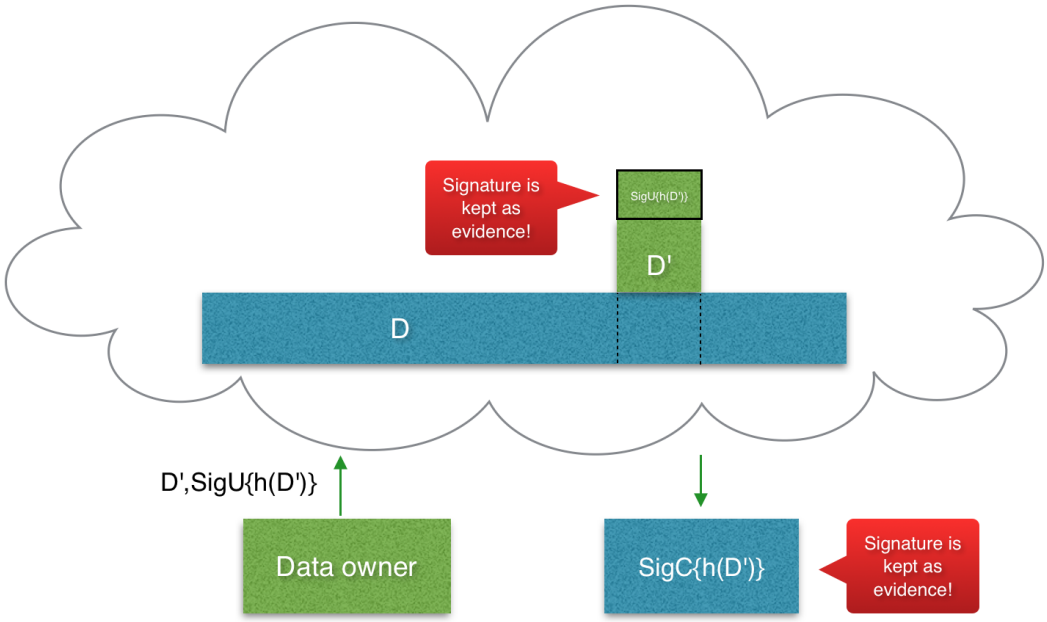

Figure 17: Model using client authentication.

A signature is also sent to the data owner providing proof that the data change has been documented. This is represented by the signature

$$\mathrm{Sig}_C\{h(D')\} \quad .$$

This cryptographic attribute is kept as evidence by the data owner in the event of a dispute. In this case it is important to note that evidentiary burden now lies with both parties and an effective model is beginning to take shape. But one more aspect must be taken into account, which is the problem of changing data that is the latest with an appropriate version parameter.

6.5.2 Verification

Figure 18 along with client authentication also a version parameter has been incorporated. As with the previous model the data owner wants to alter the last change made to

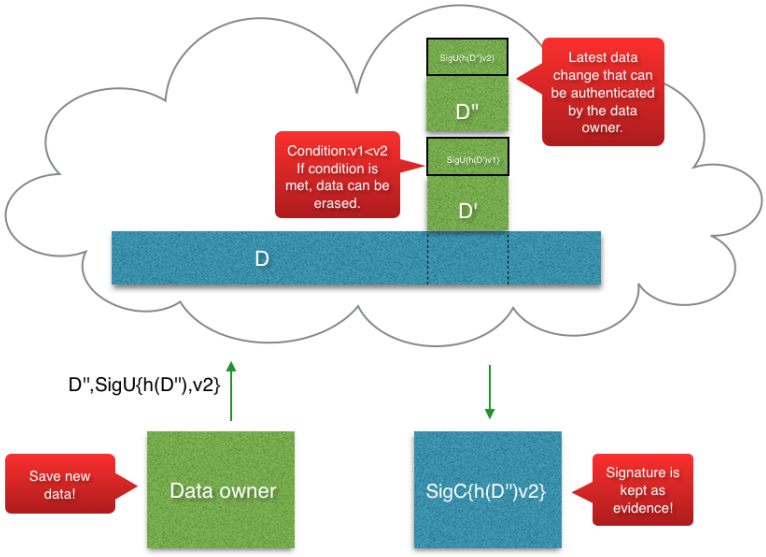

Figure 18: Dispute resolution using client authentication and version condition with multiple changes to data.

$$\mathrm{Sig}_U\{h(D')\} \ ,$$

which is now

$$\mathrm{Sig}_U\{h(D'')\} \ .$$

But how to differentiate from a new change compared to an older? This is where a version is incorporated to every future data change from the data owner by the cloud service provider. Now if we look at Figure 18, the data owner has made a change to the D', it is also represented with a version v_1. Now to alter the data the user sends data with signature

$$\mathrm{Sig}_U\{h(D''), v_2\} \ .$$

which will become the latest data saved.

$$\mathrm{Sig}_U\{h(D''), v_2\}$$

is also evidence for the cloud service provider authenticating a request to save new data with a new version. The cloud provider however has a condition that every subsequent data change must be a higher version value than the previous

$$v_1 < v_2 \ ,$$

this ensures that the latest data is saved and also that version saved is of a larger value than the previous. The cloud service provider can then send a signature

$$\mathrm{Sig}_U\{h(D'',)v_2\} \ ,$$

which represents that new data has been saved and documented. This signature is kept as a cryptographic attribute by the data owner as evidence if authentication is required in the future. In this model both parties have the possibility to provide proof that alterations to data were made.

6.5.3 Dispute Resolution

In Figure 19 a quintessence of an effective model is explained by also providing a dispute scenario. the data owner is requesting to show data that has been stored in the cloud service providers storage by providing signatures given by the cloud service provider. The data owner presents the cloud service provider with 5 signatures

$$\mathrm{Sig}_U\{h(D), v_2\},\ \mathrm{Sig}_U\{h(D'_1), v_1\}\ ,$$

$$\mathrm{Sig}_U\{h(D'_2), v_2\},\ \mathrm{Sig}_U\{h(D'_3), v_1\},\ \mathrm{Sig}_U\{h(D'_4), v_2\}\ .$$

The cloud service provider having received an authentic signature to alter changes to

$$\mathrm{Sig}_U\{h(D'_1), v_1\}, \mathrm{Sig}_U\{h(D'_3), v_1\}$$

$$\mathrm{Sig}_U\{h(D'_2), v_2\}\ \mathrm{Sig}_U\{h(D'_4), v_2\}$$

is the latest data that the cloud service provider can present to the data owner. Reluctantly the data owner wishes to see all data that correspond to signatures provided by the cloud service provider and goes to court with the matter accusing the cloud service provider of not providing data corresponding to past signatures.

The judge can request the data corresponding to past signatures from the cloud service provider

$$\mathrm{Sig}_U\{h(D), v_2\},\ \mathrm{Sig}_U\{h(D'_1), v_1\},\ \mathrm{Sig}_U\{h(D'_3), v_1\}\ .$$

The cloud service provide on the contrary can provide evidence to the judge that the changes made were made by the data owner them-self by providing signature

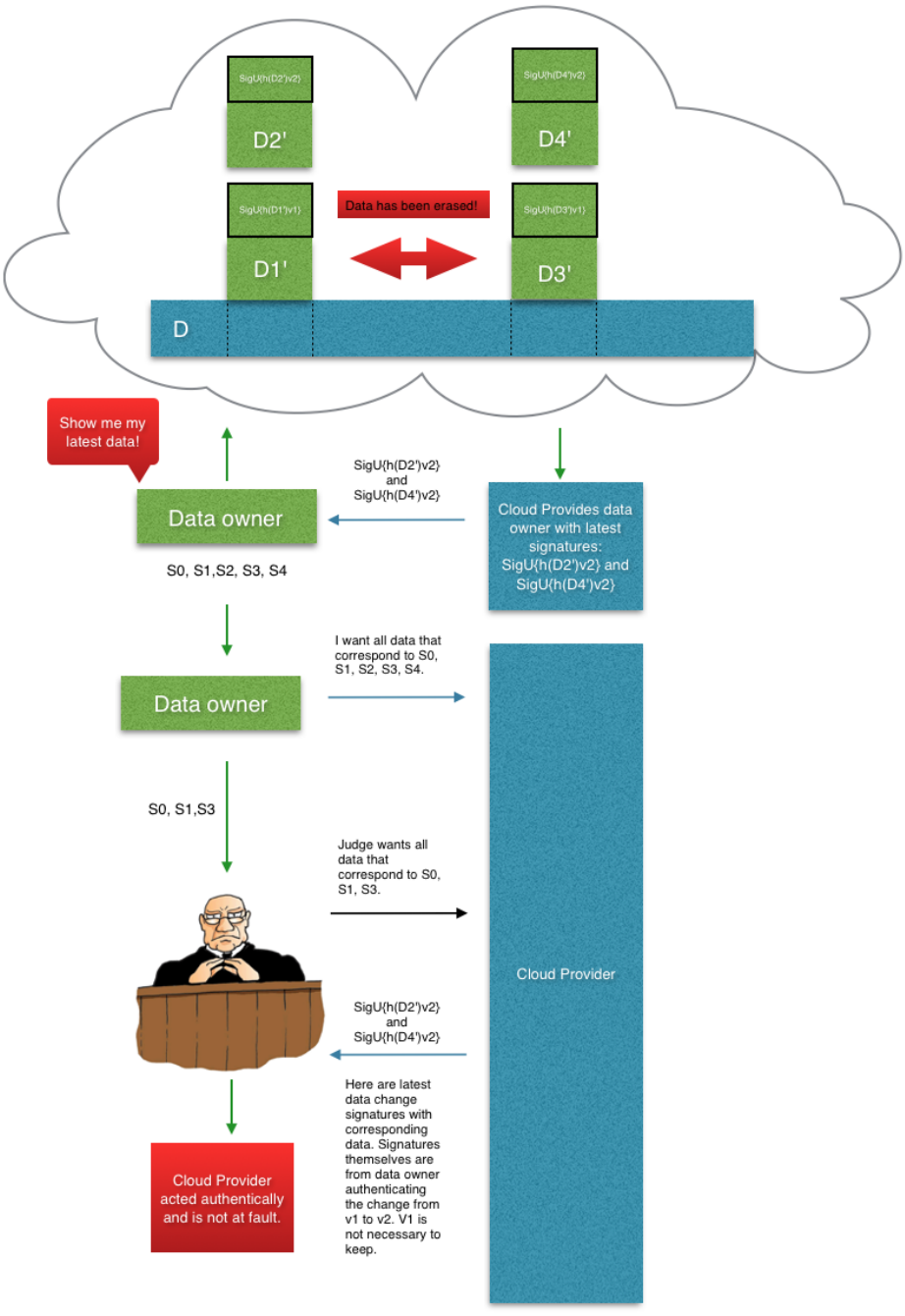

Figure 19: Dispute resolution using client authentication and version condition.

deleted the old data since

$$\mathrm{Sig}_U\{h(D_2'), v_2\}\ \mathrm{Sig}_U\{h(D_4'), v_2\}$$

which are digitally signed by an authentic user and correspond to the version condition which must be of a higher value than the previous data saved. The model works effectively for both parties giving them the opportunity to provide evidence in a court of law and can save them from compensating the other party involved. The model is also useful for the cloud service provider in the sense that storage can be effectively used for other clients.

7 Legal Background

It would seem natural that any digitally signed document has the same judicial power as a hand-written document, but no person on this Earth, perhaps Steven Hawking, can compute a digital signature by heart and it is also difficult to create a digital signature system that is completely safe to use. It is unfair to make Jane the signatory of the document responsible for every document she has ever signed in her lifetime. Then again, if John must prove the authenticity of the signature, then there would be no place in the world for digital signature. The best option is to find the best way to reduce the technological risks, define reasonable rules for solving digital signature disputes and the awareness of these rules should be clear for everyone involved in the dispute.

According to Buldas, Ansper, Roos and Willemson [8] before the notion of ”safe” digital signatures can be used- there are three main problems that must be addressed: Secure signature devices, fair rules of liability, and up-to-date public key information. The last problem has to do with efficient storage and distribution

of public key information for long-term use. In 1976 Diffie and Hellman [8] proposed the first public key distribution of an on-line telephone book however the idea was shot down by Loren Kohnfelder by which the telephone book would cause a communication bottleneck. The notion of individually signed electronic documents was far more appealing and the availability of public networks was not as developed as it is today.

The Certification Authority has a document ”The Trust Service Practice Statement”, where the rules of rules have been implemented from 2014. Here are excellent examples of which party should responsible for what in the process of signing and using Certification Authorities trust services.

The Trust Service Provider's obligations towards the subscriber[12]:

- *publish its SK PS and service-based policies and practice statements and guarantee their availability in a public data communications network.*
- *publish and meet its claims in terms and conditions for subscribers and guarantee their availability and access in a public data communications network.*
- *maintain confidentiality of the information which has come to its knowledge in the course of supplying the service and is not subject to publication.*
- *keep account of the Trust Service Tokens issued by it and their validity.*
- *inform the authorised processor of the Estonian Register of Certificates of any changes to a public key used for the provision of certification services or time-stamping services.*

- *preserve all the documentation related to Trust Services until the termination of its activity.*

- *ensure an annual audit of the information system and present the auditors report to the authorised employee of the registry to ensure continual registration at the Estonian Register of Certificates.*

From the list of obligations towards the subscribers to the service- availability and confidentiality are talked about. The ”promise” to keep the service available and confidential to the subscriber. What is interesting is that the notion of integrity is mentioned a lot. The TSP checks the integrity of cryptographic devices prior to implementation; all critical software that is installed and updated follow internal integrity procedures against viruses, malicious attacks and unauthorised software; Systems deployed are checked regarding integrity of software and that the integrity of the informations is checked before storage. The obligations for the TSP are mainly those that are put forth by the European Union Directive on digital signatures, but as mentioned before the directive was too lax in the security are and the TSP in the Republic of Estonia has made it more secure in sense of time-stamping.

The obligations from the subscriber side to the TSP[12]:

- *observe the requirements provided by SK in this SK PS and the respective service-based policies and/or practice statements.*

- *supply true and adequate information in the application for the services, and in the event of a change in the data submitted, he/she shall notify the correct data in accordance with the rules established in the service-based policies and practice statements.*

- *be aware of the fact that SK may refuse to provide the service if the Subscriber has intentionally presented false, incorrect or incomplete information in the application for the service.*

- *be solely responsible for the maintenance of his/her private key and Trust Service Tokens.*

- *the Subscriber shall use his/her private key and Trust Service Tokens in accordance with this SK PS, service-based practice statements and service terms and conditions.*

The obligations for the subscriber are quite simple- to provide truthful information about themselves or if incorrect information was provided, then to quickly update it. But the responsibility of maintaining the private key and Trust Service Tokens is still hers/his to manage.

However the TSP is not responsible for[12]:

- *the secrecy of the private keys of the Subscribers, possible misuse of the certificates or inadequate checks of the certificates or for the wrong decisions of a Relying Party or any consequences due to errors or omission in Trust Service Token validation checks.*

- *the non-performance of its obligations if such non-performance is due to faults or security problems of the Register of Certificates, the data protection supervision authority or any other public authority.*

- *non-fulfillment of the obligations arising from the SK PS if such non-fulfilment is occasioned by Force Majeure.*

As listed above the responsibilities that fall into the Subscribers breadth is keeping the their private keys secret and to use the their certificate in a manner that reduces wrong decisions on their behalf.

7.1 Certification Authority and Digital Signature Act

After regaining independence from Soviet Russia the Republic of Estonia decided to heavily develop Information Technology and network services. This led to the development of a strategy, in which a service was to use digital signatures. This idea was first proposed by our current President Toomas Hendrik Ilves, who was at the time an Ambassador for Estonia to the United States of America. Until then Estonian's have been issued ID-cards to connect them to the e-services that can be used by them [13].

The decision to use ID-cards was actually forced onto Estonian's in 2002. A citizen was required to visit the Citizenship and Immigration Authority with their passport and register the necessary information for issuance of an ID-Card. The ID- card at first was used for banking, but as time went on the amount of e-services was ever increasing that were available for use by citizens of holders of a an ID-Card [13].

The Certification Authority (Sertifitseerimiskeskus) in Estonia is a limited liability company first founded in 2017 of March 27. It is company ownership is divided into three: Swedbank and SEB bank both own a 25 percent share and 50 percent owned by AS Eesti Telekom. At the end of 2001, the Certification Authority registered themselves as a certification and time-stamping service provider in the republic of Estonia. The core services that CA provide is time-stamping, certification and digital trust services. Public organisations and businesses can benefit

from this service with secure electronic communication in day to day life. The Certification authority has a well rounded document that describes how they are able to create digital trust in electronic communication, also creating a signature that is legally binding and in accordance to the digital signatures act of Estonia. this is one of the laws that govern digital signatures, the second is the Identity Documents Actc̃itesk.

The Digital Signatures Act was first established in Estonia on December 15, 2000. It is a law that essentially governs the conditions of digital signature and digital seal use. Also the provisions for auditing the certification services and time-stamping. The second, the Identity Documents Act is a law that describes the ID-card as being main identification document and narrate requirements for documents for Estonian residents.

The Digital Signatures Act (DSA) is the first of its kind and has described in detail the legislative aspects of a digital signature and a digital seal [10]. The European Union has adopted directive 1999/93/EC ”Community Framework for Electronic signatures” which defines the requirements for digital signatures certification providers [13]. It is interesting that the European Union Directive 1999/93/EC is required by all EU states, but only Estonia has been capable of implementing it in the strictest manner. The directive was seen as too remiss, but Estonian legislation requires the certification to be valid at the time of the signing and this is one of the main reasons for the Internet at the time of the signing. The Directive does not require this and what is more interesting is that the signature in the directive is required to be generated by means that are far less secure than how Estonia is generating the signatures. This has led to a situation, where digital signatures are seen as more of a security measure than as a substitute of a handwritten signature [21].

However the 1999/93/EC ”Community Framework for Electronic signatures” has been recently replaced and updated by the eIDAS regulation [3], which stipulates the promotion for more electronic services that can be trusted. The aim is to have public organisations, businesses and citizens to use eIDAS. Previous pilots such as STORK, which is a cross border eID interoperability platform [15] SPOCS (Simple Procedures Online for Cross Border Services) is pilot for a European interoperability layer for e-government services online [14] ECODEX which is large scale pilot in the domain of e-justice, where the aim is to have cross border legal proceedings information exchanged [5] epSOS which is a project for exchanging medical information between different European Union countries seamlessly and also to reduce the amount of medical information errors [6]

All these new services/projects that were tested is the way of the future and the exchange of personal information requires the use of effective trust/integrity mechanisms to ensure authentic information is presented. These few projects though immense in volume represent the groundworks for integrity technologies.

Lets look at the definitions required to create a legally binding digital signature. A digital signature according to the DSA is [10]:

- *a digital signature is a data unit, created using a system of technical and organizational means, which a signatory uses to indicate his or her connection to a document*

- *a digital signature is created by using the data necessary for giving a signature contained in a safe signature creating device (hereinafter private key) to which the data needed for verification of the signature contained in a signature verification device (hereinafter public key) uniquely corresponds*

According to The trust Services Practice Statement the definitions brought forth are very informative and in the context have various connections to integrity [12]:

- *Digital Signature*: a data unit, created by using a system of technical and organisational means, which is used by a signatory to indicate his or her link to a document.
- *Electronic Signature*: data in electronic form which are attached to or logically associated with other electronic data and which is used by the signatory to sign.
- *Relying Party*:a recipient of a Trust Service token who acts in reliance on that Trust Service Token.
- *Sensitive Information*: information which allows for simulation or replication of service, or also for the destruction or publication of the service private key. It also includes personal information.
- *Trust Service*: described in [eIDAS] as an electronic service which is normally provided in return for remuneration and which consists of:
    1. the creation, verification, and validation of Electronic Signatures, electronic seals or electronic time-stamps, electronically registered delivery services and certificates related to these services or
    2. the creation, verification and validation of certificates for website authentication or
    3. the preservation of Electronic Signatures, seals or certificates related to these services

- *Trust Service Policy*: a set of rules that indicates the applicability of a Trust Service Token to a particular community and/or class of application with common security requirements.

- *Trust Service Practice Statement*: a statement of the practices that a TSP employs in providing a Trust Service.

- *Trust Service Provider*: an entity that provides one or more electronic Trust Services.

- *Trust Service Token*: a physical or binary (logical) object generated or issued as a result of the use of a Trust Service (e.g. certificate).

It evident that the system relies on a Public Key Infrastructure (PKI) system, where a user can sign a document using their private key and a public key can be used to verify the integrity of the signature. This verification is only to check if at the time of the signature the certificate used to sign was still certified by the Certification Authority. According to the laws and regulations passed- the digital signature used by Estonia has the same same legal power as a hand-written signature.

Among the certification services provided by the Certification Authority, among them is also a ”digital seal”. With this service, if you are the holder of the digital seal certificate can verify the integrity of an electronic document and any changes to the document can be detectable.

A digital seal by definition can [10]:

- *enable unique identification of the holder of the certificate in whose name the signature is given*;

- *enable determination of the time at which the digital seal is given*;
- *link the digital seal to the data in the document in such a manner that any subsequent change of the data or the meaning thereof is detectable*;

The use of digital seals and signatures is required by ”State and local government agencies, legal persons in public law, and persons in private law performing public law functions are required to provide access through the public data communication network to information concerning the possibilities and procedure for using digital signatures and digital seals in communication with such agencies and persons [10].

In Estonia public sector employees and representatives are by law required to use digital signature and seals to ensure the authenticity and integrity of transmitted electronic data. With the use of the signature, it can be confirmed that the document being reviewed came from a person, who is the owner of the correct certificate and use of a digital seal, it can be confirmed that the document being reviewed has not been under manipulation from the time that it was signed and to the time it is being reviewed[10].

7.2 ISKE: Estonian Information Security Standard

The IT Baseline Protection standard an information system security standard that is based on Germany’s Federal Office for Information Security (Bundesamt fr Sicherheit in der Informationstechnik). The aim on ISKE is to ensure that the electronic data in information systems has an adequate level of security. The standard is mostly for local governments and the data that they are processing, but private business owners are also able to implement ISKE security standards.

ISKE is characterized by three levels of security: low, medium and high. The level of security is determined by the data's confidentiality, integrity and time-critical availability. ISKE implementation is not a singe one-time process, it is a constant process that attempts to adapt to the constant changes in the information technology sector. The standard is updates every year according to the risks that may have emerged [16].

Regarding availability ISKE has created levels of security: K is used to characterise availability [16]:

- *K0*– less than 80 percent per year and a maximum allowable length of a single outage service work at the time of over 24 hours (ie, a single outage may have a length greater than 24 hours);
- *K1*– greater than or equal to 80 percent and less than 99 percent per year and a maximum allowed a single length of service interruption during the work up to 24 hours (ie, the length of a single outage may be in the range of less than or equal to 24 hours, and greater than 4 hours);
- *K2*– equal to or greater than 99 percent and less than 99.9 percent per annum, and the maximum allowed. The length of the interruption of a single service during working hours to 4 hours (i.e. a single outage may have a length in the range of less than or equal to 4 hours, and greater than 1 hour);
- *K3*– availability - greater than and equal to 99.9 percent per annum and a maximum allowable single outage.The length of service work at the time of 1 hour to 0 seconds (ie, a single outage may have a length less than or equal to 1 hour);

Regarding integrity ISKE [11] has defined the different levels accordingly: T is used to characterise integrity [16]:

- *T0*– source of information, modification or destruction of detectability is not important; Information is accurate, complete, and current controls are not necessary;
- *T1*– source of information, the alteration and destruction thereof shall be detectable; information accuracy, completeness, timeliness, and controls in specific cases, as appropriate;
- *T2*– the alteration and destruction thereof shall be detectable; periodic information is required accuracy, completeness and timeliness of inspections;
- *T3*– the alteration and destruction thereof shall have evidential value, information is required accuracy, completeness and timeliness of control in realtime.

Regarding confidentiality, ISKE has characterised different levels accordingly: S is used to define confidentiality[16]:

- *S0*– public information: access to information is not restricted (ie, read all interested parties, the right to amend defined by the requirements of integrity);
- *S1*– information for internal use: access to information is permitted access to the requesting party a legitimate interest in the case;
- *S2*– secret information: the use is permitted only for certain user groups, access to information is permitted access to the requesting party with a legitimate interest;

- *S3*– top secret information: Use permitted only to certain users have access to information allowed access to the requesting party with a legitimate interest [11].

What is very interesting about the ISKE standard is that there is extensive information about how to classify the electronic information, but the means to implement and how to do this is lacking. Another interesting fact about the standard is the mention of digital signatures. Digital signatures are used as a mechanism to add to the file or message to make certain the creator and if the original file is the same as the one being viewed at that point in time. This is interesting in the sense that Digital signature certificates are issued for the duration of 5 years and the fact that if for some reason, a party would like to view the authenticity of a message in say more than after 5 years, then the Certification Authority must authorize a process to view if that signatures certificate was active at the time of the signing. But how to control the integrity of the electronic data after the 5 years? one answer could be time-stamping or a certificate that lasts longer than the duration it is currently issued to Estonian citizens.

We can with ISKE define the different levels of security for an organisation, but once we have mapped the different levels, then what is the next step for the organisation to follow. ISKE is not a risk analysis mechanism, but rather an electronic information classifier.

8 Results of Analysis

Although the notion of data integrity seems simple enough, many aspects come into play when a dispute is taken as far as the court room. It is important for both

parties to have evidence of what occurred in the past to show to a judge as proof.

In conclusion of the analysis of what is integrity in the context of a third party cloud service provider is that integrity is a notion not much talked about and can be defined very differently depending on the context of which the analysis is . In this case integrity of the data protocol means to utilise technological primitives to achieve integrity for the data owner and the cloud provider. The analysis shows that to have integrity in a system the data owner who provides information to the cloud must use digital signatures to prove that an authorised person has made a request and to for the cloud provider to authenticate proof of possession. Hash functions are an effective technological primitive to use for having a verification mechanism for data to remain unchanged. Version condition must be used to by both parties to ensure that the latest changes are made and this reduces the possibility for the issue to become a dispute in court.

Theoretically data integrity in Figure 19 was achieved. The data protocol can be quite different in other IT dependent sectors like intentional insider manipulation, intentional third party manipulation, medical records, server storage, telecommunications and updating. These fields require more analysis to achieve data integrity with an effective protocol of verification and useful dispute solving rules.

References

[1] Bayer, D., Haber, S., Stornetta,W.-S.: Improving the efficiency and reliability of digital timestamping. In: Sequences II: Methods in Communication, Security, and Computer Science, pp. 329–334. Springer, Heidelberg (1993)

[2] Buldas, A., Saarepera, M. On provably secure time-stamping schemes. In ASIACRYPT 2004, LNCS 3329, pp. 500–514. 2004.

[3] ”Electronic Identification and Trust Services (eIDAS): Regulatory Environment and beyond - Communications Networks, Content and Technology - European Commission.” Communications Networks, Content and Technology. Web. 15 May 2015.

[4] Ahto Buldas, Peeter Laud, Helger Lipmaa and Jan Villemson. Time-Stamping with binary linking schemes. In Advances in Cryptology - CRYPTO’98, LNCS 1462, 486-501. Springer-Verlag, 1998

[5] ”Find out More about the E-CODEX Project.” E-CODEX: Home. Web. 15 May 2015.

[6] ”EpSOS: About EpSOS.” EpSOS: About EpSOS. Web. 15 May 2015.

[7] Buldas, A., Laur, S.: Do broken hash functions affect the security of time-stamping schemes? In: Zhou, J., Yung, M., Bao, F. (eds.) ACNS 2006. LNCS, vol. 3989, pp. 50–65. Springer, Heidelberg (2006)

[8] Buldas, A., Lipmaa, H., Schoenmakers, B.: Optimally efficient accountable time-stamping. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS 1751, pp.293–305. Springer, Heidelberg (2000)

[9] Buldas, A., Saarepera, M.: On provably secure time-stamping schemes. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 500–514. Springer, Heidelberg (2004)

[10] ”Digital Signatures Act.” Riigi Teataja. Web. 15 May 2015.

[11] ”ISKE Rakendusjuhend.” Web. 15 May 2015.

[12] ”The Trust Service Practice Statement.” Web. 15 May 2015.

[13] ”SK.” SK. Web. 15 May 2015.

[14] ”SPOCS.” SPOCS. Web. 15 May 2015.

[15] ”Stork.” Stork. Web. 15 May 2015.

[16] ”Infossteemide Turvameetmete Ssteem ISKE.” Riigi Infossteemi Amet. Web. 15 May 2015.

[17] ”http://www.thesaurus.com/browse/integrity.” Www.thesaurus.com. Www.dictionary.com, 2015. Web.

[18] ”Integrity Is What You Do When No One Is Watching; It’s Doing the Right Thing All the Time, Even...” Stampede Blue. 18 May 2011. Web. 13 May 2015.

[19] ”What Is Integrity? Definition and Meaning.” BusinessDictionary.com. Web. 13 May 2015.

[20] ”Herman Simm Phjendab Reetmist KGB Minevikuga - Eesti Uudised - Postimees.ee.” Postimees. 13 May 2008. Web. 13 May 2015.

[21] ”Digital Signatures in Estonia and the Rest of Europe ? a Look Back and Ahead.” Digital Signatures in Estonia and the Rest of Europe, a Look Back and Ahead. Web. 15 May 2015.

[22] ”What Happens When Data Gets Lost from the Cloud?” Cloud Tech News. Web. 13 May 2015.

[23] ”Patients Worried about Medical Records Going Digital.” - Amednews.com. Web. 13 May 2015.

[24] ”Data Retention.” DGs. Web. 13 May 2015.

[25] ”Car Hacking: DARPA Funded Researchers Take Control Of Toyota Prius And Ford Escape Using Laptop [VIDEO].” International Business Times. 25 July 2013. Web. 13 May 2015.

[26] ”Error Detection and Correction ADMIN Magazine.” ADMIN Magazine. Web. 13 May 2015.

[27] ”DRAM Errors in the Wild: A Large-Scale Field Study.” Web. 13 May 2015.

[28] ”Hash Function.” Hash Function. Web. 13 May 2015.

[29] Haber, S.; Stornetta, W. S. (1991). ”How to time-stamp a digital document”. Journal of Cryptology 3 (2). doi:10.1007/BF00196791

Index