SStorage

Whitepaper

Ludovic 'Archivist' Lagouardette

PUBLISHED BY NEKOIT

HTTPS://ARCHIVIST.NEKOIT.XYZ

First printing, 2019

Contents

Project presentation

1. The state of cloud storage

Nowadays, cloud storage is getting a fair amount of popularity: saving space on mobile devices, backing up important data, getting access on data while using multiple devices.

1.1 Competition

Multiple service providers, Google, Amazon, Ovh, Apple, as well as some other smaller actors have tackled the task of storing data for a variety of use-cases, pricing tables and options.

In this section you will find a report on the current state of competition. We do not aim at taking a look at the whole market but at a number of important or interesting actors.

1.1.1 Google Drive and Google Cloud Platform

Google is famous for its economical impact on the software development industry. This is also true of its Google Drive and Google Cloud Platform products.

It could be labeled the cheapest way to backup multiple terabytes[3]. You can also quote their other products Google Cloud Storage as a cheap yet very efficient tool to store data for applications and websites.

They however do not offer any kind of protection on their services, the data stored on their side is not encrypted and they may use it for advertisement purposes for example. They however are not misleading on their offer even if their product is not privacy centered at all.

1.1.2 Amazon Cloud Drive and their variety of services

It is possible to expand at length on how varied and efficient Amazon cloud storage is. They provide nearly all types of storage for any type of data structure from the typical file system to the most advanced layouts of databases.

Their prices are slightly higher than those of Google. Like Google however, they only propose encryption of the data while in transit.

Most of Amazon Cloud Service are targeted towards professional users.

1.1.3 Operation Tulip (NextCloud over Ceph)

An open-source initiative to propose a simple cloud suite with file storage and tools like a calendar and an online LibreOffice implementation.

This service is in open beta[1] and uses open source software to hold encrypted data with storage redundancy: Ceph and NextCloud.

It is not to be used for actively using the data but more as a backup solution and cold storage.

1.1.4 Backblaze

A data backup company that offers multiple solutions with diverse options. They permit the use of forms of secure encryption. They however do not encrypt all of the metadata and reserve the right to sell those to a variety of company.

They also offer a storage for live data in one of their offers. Their prizes are relatively competitive compared to Amazon Cloud Storage for example.

1.1.5 Dropbox

A well known actor in backup cloud storage system. They provide multiple tiers of pricing, from a free offer to multiple paid storage offers. All of them are meant for dead storage, for roaming and for sharing files.

1.1.6 Tarsnap

A small actor based in Canada. they offer cold storage services, encrypted and open-source on client side. They pricing is on a *"as you go"* basis, pricing network traffic as well as storage used.

The performance of the service was not tested.but from its software architecture and design, it may be relatively slow for using it as an active storage as it is delta based, as well as being unlikely to be usable with a high degree of concurrency.

1.2 Technology

Multiple technologies and their open-source counterparts can be used to handle online data storage. In this section we will explore those possibilities by comparing both commercial and free solutions where possible.

1.2.1 Google Spanner and CockroachDB

Google Spanner and CockroachDB are two database software for geo-replicated databases. They use a clock based mechanism for handling transactions, making them as fast as their clock synchronization. CockroachDB have however lower requirements on clock accuracy that Google Spanner does[5].

[1] can be tested by anyone

Google Spanner is a proprietary product from Google. CockroachDB is an open-source project from CockroachLabs made to implement as much of Google Spanner features as possible. It also intends to try to be compatible with PostgreSQL to ease application porting[6].

Both of those tools can be used to implement either a block based storage or an object storage usable to implement a geo-replicated filesystem.

Using CockroachDB as a back-end to implement SStorage was envisioned, but latency tests made us choose to use a custom implemented data server. SStorage uses a similar way of resolving database conflict (see the sequence diagrams **??** and **??** at page **??** and **??**).

1.2.2 Ceph, RADOS and CRUSH

Ceph is a distributed data storage system. It uses the RADOS (Reliable Autonomic Distributed Object Store), a storage system designed around the idea of placing data in predictable place following a mathematical equation. This is named CRUSH, for Controlled Replication Under Scalable Hashing.

Placement of data in SStorage system follows some concepts from Ceph, RADOS and CRUSH.

Ceph is currently in development by the CERN. Ceph is used as a back-end for many types of storage, from filesystems to block devices and storage for scientific data.

As mentioned in the listing of other actors, the Operation Tulip project are using it to store the files they manipulate. Other not mentioned actors like Ovh use it for storing Virtual Machines and as storage for cloud computing for example.

1.2.3 NextCloud

NextCloud is an open-source system written in PHP to be used as a front-end for cloud hosting. It supports WebDAV and other protocols as well as providing multiple productivity features like text edition, spreadsheets and calendars.

It is slow due to having been designed in a programming language unsuitable for performance applications.

It supports end to end encryption.

1.3 Hardware and hosting

Naming it cloud storage doesn't mean the data is in some phantasmagorical place. As such we will study here the possibilities for one to deploy their own cluster of servers to host their own data.

For that will be provided a comparison of pricing of the hardware required to deploy their own solution online for a data size around 50TB($\pm$5%) of storage.

As the table Table 1.1 expresses, buying hardware in 2019 is not as efficient as renting servers if you do not have the ability to host your servers in your own premises.

1.3.1 Brand new hardware

Brand new hardware is generally a real investment for an individual or a new company. It is also a technical choice that can have lifelong consequences on the business as computers are subdivided in families that each have specific features and behaviours.

System	Upfront price	Price per GB per year	Ac. D.[2]
SuperMicro SC825TQ-560LP ×3 and SuperMicro 5018D-MF (new)	USD15100 +USD900/m	USD0.21	5
HP ProLiant DL180-G5 ×4 (refurbished)	USD3350 +USD900/m	USD0.21	3
Ovh rented servers ×4	USD780/m	USD0.20	0

It is to be noted that the performance is also decreasing with each category down, as the upfront price is rising. Monthly fees for non-rented items are the housing for the hardware.

Table 1.1: Server pricing

Of those families, named architectures, we will consider two: the `x86_64`, also referred as `amd64`; and the most recent architecture from the `ARM` group, the `ARMv8` architecture and its variants.

Both of them share the minimal set of features for a type of storage named a `memory mapped hash table` to be implementable.

`x86_64` architecture

This architecture is common to most modern computers, laptops, workstations and servers. It is therefore easy to make software for it as it is well documented.

It have the huge drawback of being power hungry, having been extended over time, it tends to be consume more power and require proportional cooling.

Taking for example a server from SuperMicro SC825TQ-560LP, an estimate of price around 4'500USD per server three times for the data storage, as well as any other server for handling coordination of data storage.

For coordination, a server of the likes of a SuperMicro 5018D-MF, for which we estimate a price of about 1'600USD equipped with a proper network card for handling connections to the storage servers properly is appropriate.

`ARMv8` architecture

This architecture being new, it is hard to adventure into pricing it, but adapted servers for storage equipped with ThunderX2 CPUs from Cavium would do well as storage server and likewise equipped servers with a ThunderX2 adapted for computationally heavy loads would fit the use case as a coordination server.

This setup is however untested and it would not be possible at the time of redaction of these lines to test it for us. These servers also may not run some operating systems critical for safety of network infrastructure like OpenBSD as of the writing of these lines[3].

1.3.2 Refurbished hardware

We looked into the products of professionals in sales of refurbished hardware. It is advised for those with small budget a constellation of HP ProLiant DL180-G5, with a price per server of about 950USD (disks being new), and any server with a decent enough set of network connectivity to not be a bottleneck in coordination.

[3] last verification on January 14th 2020

1.3.3 Rented dedicated servers

As for dedicated servers, Ovh proposes to rent servers for 230USD per month per server with an added 80USD per month for the coordination server. This doesn't encompasses any backup server additionally needed to guarantee fast replication if one of the three servers fails, but it takes into account all hosting costs.

1.4 The users

We conducted a survey on Telegram and Discord about privacy in cloud storage and cloud storage usage. While with hindsight, some questions given the population surveyed lead to unsurprising responses (Telegram being quite security oriented), they may be interesting on certain regards.

The survey was conducted on 23 people, most of them from computer science and programming related groups on Telegram and politics and economics related groups on Discord.

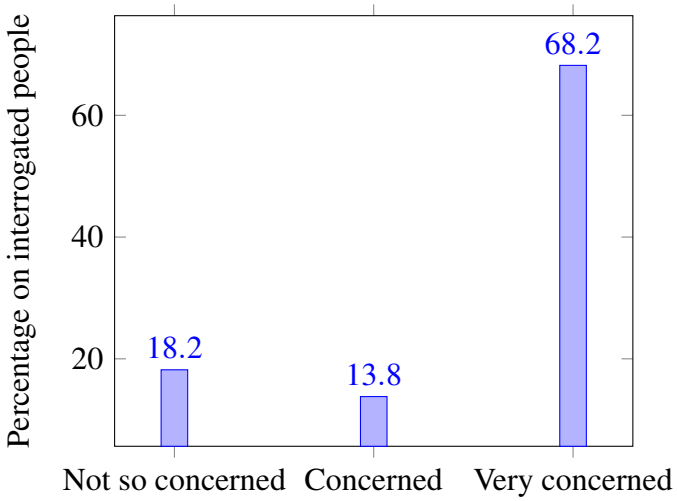
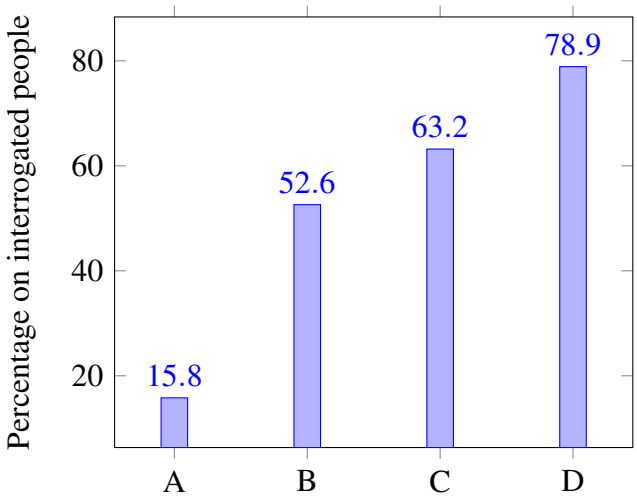


Figure 1.1: Concerns about privacy

First of all, most of the people interrogated were Telegram users, in the demographic of Telegram users 53.33% are using a VPN, and 50.00% or the surveyed people that do not use Telegram use a VPN.

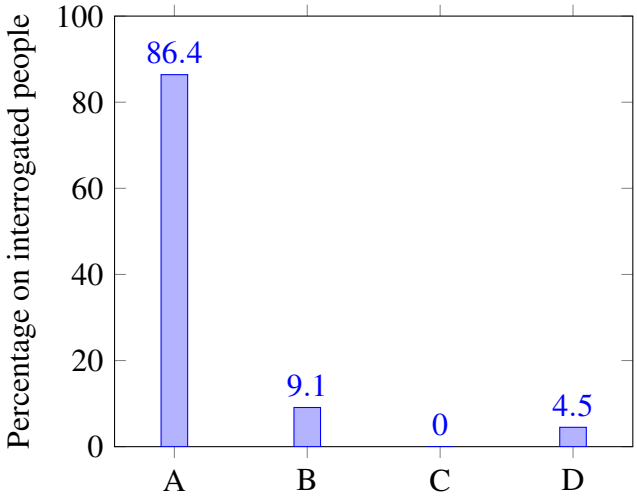
It also shows that the privacy concerned population have significantly more trust in open-source community approved cryptography than in government approved cryptography.

Lots of people also use encrypted hard-drives but most are cold to the use of cloud storage. From a few interviews most of them harbor distrust of the majors cloud storage service providers. Our goal is to provide a solution that these people can trust to store their hot and cold data.



- A: Harddrive encryption (Hardware or commercial solution)
- B: Harddrive encryption (Open-source solution)
- C: VPN
- D: Telegram

Figure 1.2: Use of privacy enabling tools



- A: Open-source community approved cryptography
- B: Government approved cryptography
- C: Hardware implemented cryptography
- D: I don't know

Figure 1.3: Opinions on cryptography

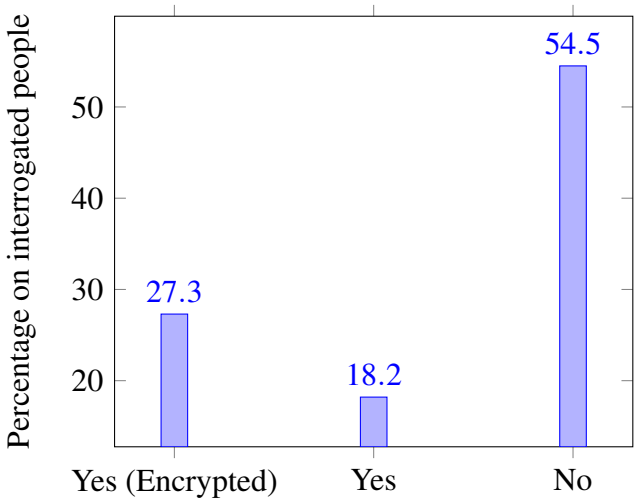


Figure 1.4: Use of private online data storage

2. A personal view on privacy

This chapter expresses the view of the author of both this documentation and the software associated with it and only of him.

Privacy is a daily concern. Everyday people use objects made to guarantee some levels of it, from curtains to acoustic insulation, from locked doors to security cabinets, privacy is something that concerns doctors, lawyers, engineers, inventors, chefs, military staff… But also each and everyone to some degree.

Sometime privacy is an indirect concern: an archival company should not take a peek at your doctor's or lawyer's files and cases. Sometimes such an indirect concern reaches to be a concern of someone through friends, business partners, lovers… You would not want someone to learn your friend's secrets through you.

50 years ago, someone wanting to learn your secrets had to listen on your telephone line, breach in your office and open your safe, go in your house or hire a detective. Today, that person may just be able to buy your secrets.

In this chapter we will talk about a variety of topics related to digital privacy, from its premises to its implementation.

2.1 On terms of service

In the software and service industry, terms and conditions of service are the typical way for a company to announce the type of data they collect and the use they make of said data.

Those conditions may also tell a person to who the data sent on their service belongs, some services taking property of, for example, all and any picture uploaded to them.

This is in my opinion problematic from a moral standpoint when it is not explicit that the service acquires your information with your consent but on terms you may not entirely agree with for the simple reason that those terms are buried into a huge quantity of legal information.

The projection of that issue is when the very same terms and conditions allow for the company to sell or provide the information, generally non-anonymized, to a third party

without additional demand for consent. This is extremely common in companies that offer services "for free" or for very low prices compared to the cost of the actual service.

2.2 On advertisement

Advertisement is a very close issue to the one above. Most advertisements online run code on the computer than sees the advertisement to ensure the advertisement is seen by a human and not a computer. The advertisement also collect information to uniquely identify the user and link the user to the data in the page and website the user is visiting. This allows the advertisement company to run finely adapted advertisement.

One of the bad consequence of that is what we saw during the Cambridge Analytica incident in 2017, when the company of the same name got tasked to influence voters of the United States of America targeting them specifically on points of the opposing party they were likely not agreeing on with advertisement and viral videos.

In that very scandal, it had appeared the developed database could for example accurately point out users that were in favour of free access to guns[1].

Those are however not the average practices. For example the DuckDuckGo search engine providers only provide advertisement depending on the exact query entered and nothing more, not collecting any data. On a grayer side, Twitter provide access to tweet statistics and advertise this feature to all users, letting you know how they get their money and offering you to go from user to consumer very easily, which is a better practice than silently collecting data for sales to companies only.

2.3 On manipulation and political acts

As technology evolved, we gained power to make links on multiple pieces of data about users like presented in the previous part of this chapter, we also showed how data collection can be used to further a political agenda with targeted advertisement. But this omit the most clear and easily forgotten form of political warfare, collecting the other side's information at the source. This also applies to people that may want to blackmail someone else or just ruin their reputation for hidden motives or just the challenge of it.

Numerous times have we saw such breaches. From the Watergate scandal half a century ago to Apple iCloud breaches in 2014[4], stealing data is a typical way of spying on your opponent whatever the game being it political, gambling, contests, art, etc... It can also be used to obtain an unfair advantage in trials and other circumstances.

We here see the importance of privacy for public figures just as we saw it for voters in the last part. It goes the same for other methods like viral advertising and scientific cherry-picking when pushing a political agenda.

It is also valid on a bigger scale like for example, standardization of encryption by a country in order to enforce it to be breakable like it happened in 1977 with the DES 56 encryption primitive[8].

2.4 On misrepresentation of encryption

If you want to further your understanding of encryption before proceeding, I advise you to take a read at the Appendix A

Encryption is often misrepresented, both by lots of governmental figures and by lots of commercial software providers.

Nowadays, most of the web communications are encrypted and at least partially authenticated. Authentication is done through asymmetrical encryption based systems, contacting an intermediate named a certificate provider. Some of the certificates are included in most mainstream browsers (for example, the certificates of `google.com` are embedded in Android devices and in Google Chrome), which secures the communication with those entities if the private key is not compromised.

This doesn't mean that any data sent to those services is encrypted once stored on the provider: most providers do not store data encrypted as it brings computing costs up by a very significant margin if they need to access that data.

Similarly, it is considered bad practice to store passwords in a readable format, to protect them, specific cryptographic techniques exist so that it is possible to verify a password from a form of said password transformed with a one way transformation named a cryptographic hash function. That being said, some companies still store password in readable form in their databases.

This means that, access should be compromised on the database of a company or within any vulnerable part of their computer system, data could be entirely compromised. This has happened a lot in recent years, and is bound to be a phenomena that multiplies should companies not start caring for their customer's privacy.

Such compromise can happen in various ways, and having physical access to the machine makes it easy to access most if not all data on the machine. Most companies renting their disk space, servers, or computing power from other companies, it means you rely on companies contractors not to sell your data per the terms of their hosting services too, and encryption of the transit of data will not protect you from this issue.

In the meanwhile, all companies play the game of demagogy and present themselves as perfectly secure. Some have the transparency to present you the way their technology works, relying on open-source software to provide their services.

There is also the topic of back-doors in those services for governmental checks. The main issue with them is the following: if the government can access your data without you knowing, then virtually anyone can do the same. It adds a critical point of failure in the system. It also means the service is not usable for sensitive topics like defense and military uses.

On an even worrisome topic, some companies boast to feature encryption of user data, while they only ever ensure this encryption on transit, or advertise it while not all of their offers are actually featuring encryption of user data.

3. Izaro storage

The way we decided to implement our storage focuses on protection, obfuscation and performance. We will here explain the influences and consequences of these ideas.

3.1 Goals

We want to provide an online storage with the following properties:

First of all, it must be georeplicated. It is not okay to lose service access due to the loss of one server farm on our own side.

Then, the data must be protected, we ourselves should be entirely unable to read it, we should also be unable to read the metadata that is not absolutely required to provide the service.

Also, any part of the data must be fast enough to access that it is hard to differentiate our service from access of an encrypted hard-drive given a good enough network connection, same goes for writing.

Finally, it must be flexible and adaptable to multiple use-cases.

This leads us to the following idea: we are aiming to create a service that can store encrypted data, it must be able to store it in a layout similar to a disk, this way it possesses the same capabilities as a hard drive disk. The key to decipher the data is stored online but encrypted using a key derived from the user password. Authentication requires the user to be able to read the password to get a token. It is possible to leave said token disabled and enable it only with a second authentication factor.

We want our system to be protected from the point of view of our customers, as such, we aim at it having a code-base readable and short enough to be explored completely in 3 days by a developer with access to enough documentation.

3.2 Principles

Our project aim to follow the following principles:

- Principle of least knowledge: if it is possible for us to never have access to a readable form of some data, then we should not make it mandatory or provide alternatives.
- Principle of greater usage: if it is possible, we have to use the most out of the algorithms we use, be it cryptographic primitives or other algorithms.
- Principle of openness: we aim to disclose any incident that may happen, and to disclose any request by officials to access anyone's data.

The principle of least knowledge is upheld in the very design of the system: only the user can make sense of the address space of both the file system and block device. To provide an analogy, the data is stored in multiple boxes. The user side software randomly labels the boxes and seal them (that seal is the encryption). If you store data that overflows from one box, you will store in multiple boxes. deciphering any data requires to know which box is the first one and which is the next one, but that very piece of information is not stored on the server: it is stored in one of the sealed boxes.

Furthermore, the labels of each block of data can be used as a piece of the encryption process, this is an example of the principle of greater usage: any additional information that can help make the system safer, we will use it.

As for the openness principle, it is just as stated, we will disclose any demand that are made as soon as they are made as well as our responses to them. We will disclose any security issue or concern we receive. We will provide tools for anyone to be informed of these information through multiple channels.

3.3 Consequences of those principles in the design

The first consequence of that design is that it is impossible for us to decipher data sent to us. We however had a trade-off to make to maintain a healthy performance for reading or writing sequential data of considerable size: if a big file is written at once and the client send the data in order, the big file will be stored approximately sequentially in our database. This can be however be blurred by not writing the big file in order.

Another consequence is that the most permissions that can be handled on a block (a unit of a file system) is making it either non-readable, read-only, or write, and that this unit of file-system is not suitable to implement in system access control (for example, Microsoft Windows permissions, Linux system permissions...), meaning those are to be enforced by the client computer. Whatever the permissions, this means that having any file-system write permission allows a user to perform any operation on the file-system as a whole.

3.4 Data life cycle

Here is a list of the data that may be collected by us in any interaction with our software. This data is sorted by interaction.

User action	Data collected	Reason
Account creation	Email	Authentication
	Nickname	Authentication
	Password (Obfuscated)	Authentication
	Time of creation	Bookkeeping
Connection	Connection time	Authentication
Payment (below 20€)	Amount	Accounting
	Time of payment	Accounting
	Type of payment	Accounting
Payment (above 20€)	Amount	Accounting
	Time of payment	Accounting/Bookkeeping
	Type of payment	Accounting
	Invoice address	Accounting
Writing data	Server time	Data protection (Consensus system)
	Number of used blocks	Accounting/Bookkeeping

Table 3.1: Table of collected data

4. A personal view on business practices

This chapter expresses the view of the author of both this documentation and the software associated with it and only of him.

On the internet, we encounter a huge variety of businesses and of company cultures. We also encounter a just as huge variety of bad business practices.

As a person with peculiar ways to see good and bad, I will say that is actively doing things that may be hurtful to a customer or have dire consequences to them as bad.

4.1 On selling the user

For some companies, users are the resource that they sell to gain their money. I would like to put down 3 forms of business just after that sentence:

1. Companies that sell products and services
2. Companies that showcase or regroup other products and services of other companies and sell them
3. Companies that provide a service to someone for free in exchange for their personal information in order to sell said information to other companies

I think it is fairly easy to see how a user could be troubled by the idea that the service they are providing them is a honeypot for advertisement information. I make a difference between that and providing advertisement to user of one's platform in the idea that they are providing information to their users (advertisements) but not selling the information of any individual user of their platform, putting it akin to advertisements on television or on cars.

I see two issues in selling personal information from users *even if they signed a contract to agree you do* and first is that users do not know to who their personal information has been exchanged with; in short, they lose control of their privacy. Coming from a country where keeping video protection records for more than necessary is illegal and where one can forbid companies from using their phone number for any kind of communications, I was raised in the idea you are free to exchange your own personal information but not anyone's else.

This leaves you master of your information at all times.

The second problem is a moral one, and it is double. First, their user is a product, no longer so much of a human. This means that more than respecting them as individuals, you have to get a lot of them to create a significant revenue. Then, it is about how explicit companies that do that kind of model inform their users of the way their model works and on the alternatives to having your data sold they do provide, which are generally nonexistent.

I do understand people would be willing to sell their data for a service free of charges, It should be understandable people would also be willing to have that service and pay for it to be in a situation where their data is not sold to third parties.

4.2 On misrepresenting the invisible

Some companies are proud to showcase their service/products as a first class services/products, whenever it is customer support or products, while it is actually third-world exploitation or exploiting illegal immigrants[7].

No need to present any details on morals implications on lies and of employing low qualification labor for as cheap as it is possible. I will focus on how bad that is for your customers. First of all, under-qualified labor is very likely to provide wrong advice for products or to be very under performing providing a service, not to mention building or assembling any sort of products.

Some also present their products like they are so advanced they are magical while they actually provide very few added value.

On this side is also the representation of the mythical beast named the Cloud. People tend to represent it to themselves as their data stored in some imaginary location, as it effectively can be anywhere in the world. Where a company could make sure data about a single customer is in a handful of their data centers so that they could say "your data is in located in our datacenter of *name of the city* in *name of the country*" it would make for a nice improvement.

4.3 On practice of transparent business

I think any level of openness of practices in a business is good. When you show the insides of your business and of how it works, I think it both bears trust and customers to you.

Companies with open business practices have in my opinion the greatest chance of surviving for a long time and to avoid scandal. I think it can be as open as presenting the cost of the products sliced by its components and margin to the customers. This brings a relationship of trust between the business and its customer, making the business more capable of actually changing the price of their products given changes of the cost they incur to make them.

II Annexes

A. Encryption popularized

Encryption: The act to transform a message into a random looking cipher-text. The original message is often named the plain-text.
Cipher: The mathematical function that transforms a plain-text into a cipher-text

Encrypting data can be done in various ways. Each way have its properties and its resistances to certain types of attacks. All modern cryptography is key-based cryptography. It means that the way we encrypt data is not secret, what is secret is a value, named the key, that is used to encrypt the data.

A.1 Properties of encryption

We will here explain some of the properties a cipher can hold.

A.1.1 Resistance

A cipher generally have its resistance expressed as a power of two (e.g.: 2^{103}) or as a number of bits of entropy (e.g.: 103 *bits*). It is to be noted that this scale is not linear: it is exponential.

This means that a cipher that have 104 *bits* of entropy is 2 times harder to break than one with a resistance of 103 *bits* of the same family. Comparing resistance between different families is not relevant.

A.1.2 Compactness

Compactness of a cipher means that if you encrypt a message of side n you will obtain a cipher-text of the same size. Conversely, If a cipher can generates a longer cipher-text than its message, it is said to be not compact.

for example, let's consider a simple cipher: for a message A, read it as a number and multiply it with a value that will be the key.

If your message is for example 8 digits, like 00005555 and the key is 12345678, the cipher-text will be equal to $5555 \times 12345678 = 68580241290$ which make a 11 digits cipher-text from a 8 digit message, and hence make the transformation non compact.

An example of compact transformation is the truncated addition, also named modulo addition, used by systems like one-time pads.

A.1.3 Forward secrecy

Forward secrecy is the property of a encryption system to protect parts of the messages given some were compromised. The typical use-case is, for example, to prevent decryption of a message if the message before that was compromised.

Forward secrecy is very important for messaging systems as it is suited to the fact that the key may be changed quite often, or that the key is not sufficient to break the encryption.

A.1.4 Durability of secrecy

Durability of an encryption system depends on two main factors: the resistance, all flaws taken in consideration, of the encryption scheme and the evolution of computers and their accessibility in the future.

You can put it in the terms of encryption having a expiration date. Past this time, someone that started decryption of the data you encrypted immediately may have deciphered your data.

A.1.5 Encryption flaws/Cryptanalysis

Some encryption systems have know flaws. Flaws in cryptography can either render the encryption obsolete or lower the work required to find a key. They always affect the durability of the secrecy in different scales.

For example, some flaws of the AES algorithm make it less safe by an order of magnitude of several thousands times faster to break. A complete brute-force attach is believed to take 3×10^{51} years at most for the AES-256 variant with 50 surrealistic supercomputers able to compute a billion billion keys per second, With that it is clear that breaking it even with an advantage due to flaws is not realistic in a human lifetime.

Some other flaws may compromise systems are flaws that compromise completely cryptographic systems or that are predicted to theoretically compromise it in the years to come. An example of that is prime number factorization based asymmetrical, currently threatened by quantum based cryptography.

Research of flaws in cryptographic systems is named cryptanalysis.

A.1.6 Side channel attacks

Side-channel attacks refer to attacks on a cryptographic system that do not affect the way the system is designed but the way it is implemented. For example, using the sound made by electric current in a CPU have been used against some implementations of the OpenSSL library to deduce part of the key that was being used.

It is very hard to predict side channel attacks, and just as hard to prevent them.

A typical mitigation is for example to ensure all cryptographic operations take a constant amount of time. This prevents a typical attack called a time-based side channel attack.

A.1.7 Homomorphism

Homomorphism means that for a message A and an operation $f : x$ (for example, if $f : x \rightarrow x \times 2$ means the operation of multiplying by 2), if you apply a cipher to A and get a cipher-text B, there exist a way to apply $f : x$ to B in such a way that decipherion of B gives you the result of applying $f : x$ to A.

Expressed more simply, it means the you can execute operations on encrypted data without requiring to decipher it or understand it. Very few encryption mechanisms are fully homomorphic and those are mostly in research[2].

A.2 Types of encryption

Encryption can express itself in different forms regarding to its way to handle the cryptographic key. Some have only one key, that must be known for encrypting and deciphering the data, we call those symmetrical ciphers; some have two keys, one for encrypting and one for deciphering, we call those asymmetrical ciphers.

A.2.1 Symmetrical encryption

Symmetrical encryption aims to encrypt data on a two way channel. The key allows you to both write encrypted data and read encrypted data, making it very suitable for securing a network channel once the keys have been safely exchanged, or to encrypt a disk.

- Rijndael
- Chacha20
- Blowfish
- Serpent
- Twofish
- CAST5
- RC4 and RC6
- DES
- 3DES
- Skipjack
- IDEA

Figure A.1: List of symmetrical ciphers (non-exhaustive)

Symmetrical encryption also have the advantage that in some cases it is possible to interweave some amounts of the encrypted data together, making the data harder to decipher if a part of it is missing. This is especially used when encrypting data that is read sequentially like network data, but this feature is not suitable for encrypting data on a disk or on any random access media.

A.2.2 Asymmetrical encryption

The goal of asymmetrical encryption is to provide ways to authenticate messages, ways to encrypt a message with a key and decipher it with a different key, and more generally ways to exchange secrets.

- Prime factorization based (RSA)
- Elliptic curve based (ECDSA)
- Paillier crypto system
- Lattice based (NTRU, BLISS[2])

Figure A.2: List of asymmetrical ciphers (non-exhaustive)

It is evolving a lot nowadays as the most used algorithms are not extremely resistant to being deciphered by quantum mechanics based computers, in particular, systems based on the prime factor decomposition problem are hard to solve by typical computers but in theory not as hard by quantum computers.

This kind of cryptographic systems are generally used to exchange keys for symmetrical encryption.

A.2.3 One-Time Pads

One time pads are a cryptography technique that suppose both sides of a communication own a shared pad at least the size of the data to encrypt. Elements of the pad are added with a modulo addition to the plain-text to generate the cipher-text.

There is no known way to decipher the data without the pad, making One-Time Pad (or OTP, not to be mixed with One-Time Passwords) the safest cryptographic scheme, albeit an unrealistic one for large amounts of data.

It also relies on a different channel to transmit the pad in case of communications, yet doesn't in the case of storage.

Bibliography

[1] France 2. *Facebook, l'envers du réseau*. TV. Apr. 2018. URL: https://www.youtube.com/watch?v=9kpKDaF3IFw (cited on page 12).

[2] Craig Gentry. "A Fully Homomorphic Encryption Scheme". AAI3382729. PhD thesis. Stanford, CA, USA, 2009. ISBN: 978-1-109-44450-6 (cited on pages 25, 26).

[3] Linus Media Group. *I Hope Google Doesn't Ban Us... - Abusing Unlimited Google Drive*. Youtube. Aug. 2018. URL: https://www.youtube.com/watch?v=y2F0wjoKEhg (cited on page 3).

[4] Leo Kelion. "Apple toughens iCloud security after celebrity breach". In: *BBC* (July 2014). URL: https://www.bbc.com/news/technology-29237469 (cited on page 12).

[5] Spencer Kimball. "Living without atomic clocks". In: *Cockroach Labs Blog* (Feb. 2016). URL: https://www.cockroachlabs.com/blog/living-without-atomic-clocks/ (cited on page 4).

[6] Raphael 'kena' Poss. "Why CockroachDB and PostgreSQL Are Compatible". In: *Cockroach Labs Blog* (Aug. 2018). URL: https://www.cockroachlabs.com/blog/why-postgres/ (cited on page 5).

[7] Louis Rossmann. *Samsung one-ups Apple with sweatshop labor: contractor raided by ICE*. Youtube. Apr. 2019. URL: https://www.youtube.com/watch?v=XjW2tMeO3Mk (cited on page 20).

[8] Wikipedia. *Data Encryption Standard — Wikipedia, The Free Encyclopedia*. [Online; accessed 30-July-2019]. 2019. URL: http://en.wikipedia.org/w/index.php?title=Data%5C%20Encryption%5C%20Standard&oldid=905592598 (cited on page 12).

List of Tables

List of Figures