University of Glasgow | School of Computing Science

Building Applications on the SAFE Network

David Brown

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — February 25, 2018

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

> The SAFE Network is a decentralized data storage and communications network that provides a secure, efficient and low-cost infrastructure for everyone [1].

The SAFE Network [1] is an open-source project being developed by a Scottish company Maidsafe [2]. Their aim is to build "The World's First Autonomous Data Network". An 'Autonomous Data Network' in simple terms, is "...a network that manages all our data and communications without any human intervention and without intermediaries" [3]. This network will be decentralised, splitting data and then storing it around the world on computers called 'vaults'. I will go into more detail on the SAFE Network and how it works in a later chapter.

1.1 Aims

My main goal for this project is to explore the usage of the SAFE Network for the purposes of providing decentralised and 'permissionless' access to websites like Wikipedia.

The program I build will be able to upload ZIM[4] files to the SAFE Network, provide public links to everyone using the application and then be able to read/browse the files. These files will be immutable, this is to help ensure that once a user has uploaded a ZIM file to the network, that it cannot be altered by anyone once it has been uploaded (apart from the functionality to delete it). ZIM files are a convenient way of being able to package/archive a website (web content) into an offline file that can then be browsed and distributed easily.

By building this application, it will be easy to draw conclusions on whether this method of 'archiving' websites to the SAFE Network is sensible. ZIM files themselves can easily be tens of gigabytes in size. Thus in building this program it will be feasible to ascertain how well the SAFE Network can handle large files.

1.2 Motivation

In my opinion, the right to liberty and the unobstructed access to information is the most important right we have. Throughout history, a common tactic of *evil* governments or people is to block access to information. By doing this, they try to break down a culture, to control people. The most prominent example of this was the Nazi Book Burning Campaign [5]. The goal of this was to destroy any literature or information that could subvert the ideologies that Nazism is built upon. A modern day example of this is North Korea, government censorship

infiltrates every aspect of the culture [6]. State run media spouts constant lies and propaganda to try and silence any doubt that their way of life is entirely controlled and orchestrated at the whim of the countries dictators.

Close to the heart of this project, is the block on Wikipedia in Turkey [7]. One finds it hard to comprehend that a country, enjoyed by many as a popular holiday destination, is so publicly blocking their citizens access to freedom of speech and educational content. It is for the above reasons that I find the promises of the SAFE Network so alluring. An 'internet' in which it is impossible to block or orchestrate what content can be accessed without blocking access to the entire thing. A network that is built from the ground up to protect the free access to data, working autonomously and without judgement on what information is being stored and shared. By hosting archives/copies of websites such as Wikipedia and Wikispecies on the SAFE Network, it could allow people from all over the world to access content that they haven't before. More importantly for some people, for them to be able to access the content without their governments being able to detect what they are looking at.

Chapter 2

The SAFE Network

2.1 Vaults and Clients

The core aim of the SAFE Network is to replace the 'Internet' as we know it today. The internet has grown since its inception into a living, breathing organism thats growth never seams to stifle. The internet is a highly curated and structured system, Routing around the network is not always 'shortest path first'. Things are more complex than that. Governmental policy especially can have a large impact on how people interact with the network, whether that be Turkey blocking Wikipedia or the US abandoning Net Neutrality. This area is where the SAFE Network starts to deviate greatly from the *traditional* internet. The SAFE Network is a 'Autonomous Data Network'. The autonomy characteristic of the network is one of the most important features of the network and is what I believe to be its biggest asset.

The SAFE Network is comprised of *vaults*. A Vault is a singular program/application that a user runs on their computer, whether that be a server or a Raspberry Pi. A vault is given a set amount of storage by the user and then it uses this allocated storage to 'farm' data. In order for a given vault to join the network, it must pass a 'Proof of Resource'. This initial *test* is used to validate that the vault has enough bandwidth and CPU power to be able to adequately perform its job. A vault *farming* data is analogous to a *miner* in the Cryptocurrency world. A *miner* is fundamentally there to secure the network, they are rewarded for doing this by the allocation of newly minted coins of the given Cryptocurrency (usually). Where a miner *mines* for new coins, a farmer *farms* data. Similar to how a real world farmer looks after their crop/animals, a farmer/vault on the SAFE Network looks after data. Understanding that nomenclature is quite useful in understanding the function a vault/farmer. Once a given vault is successfully storing data, it is rewarded with Safecoin. Safecoin is the cryptocurrency of the SAFE Network, it is earned by farmers and spent by uploading data to the network. The expectation is that as the cost of CPU/Storage falls with time, the value of the Safecoin will increase. As in, the amount of raw bytes that a given Safecoin would allow the storage of increases. To interact with the SAFE Network a user doesn't need a vault, by using a client a user can interact with the network freely. Consuming data from the network doesn't cost a client/user anything, it is only when they want to store data that they need an available balance of Safecoin to do so. A user can have a secure connection to the SAFE Network through the secure routing layer Crust, which I will speak about later. This secure connection hides a users IP address from the network as a whole and ensures the safe transmit of data.

The only time a user that owns a vault interacts with it, is on startup. To allocate the storage etc. This is the autonomy of the network. Once vaults are created and start communicating with each each other there is no intervention by humans. The network itself decides where data is stored, the price that a given amount of data costs, how to route data etc. In order to achieve this autonomy, the network has a vast amount of redundancy built into it to help secure data. Meaning you can lose several vaults from a large enough network size and not

be in any risk of losing data.

2.2 The Architecture of a Vault

Data that is stored on the SAFE Network is split up into small 1Mb chunks. I will explain this in depth later but the core idea is that each 1Mb chunk of data. Each 1Mb chunk of data is then hashed to give a unique address in 256-Bit XOR Address Space. This address is then used to determine where that piece of data will be stored. This mechanism helps to mitigate data being duplicated on the network, two identical pieces of data will have the same hash and therefor will only be stored once. Maidsafe's innovation was in the creation of what are called, 'Disjoint Sections'. These *sections* are groups of vaults that are responsible for a certain range of the 256-Bit XOR Address Space. By default, the network requires a minimum number of vaults to sustain the network. At the time of writing this is 8 vaults. These 8 vaults form a complete 'section' and are responsible for the storage of the entire 256-Bit address range. As more vaults join the network, this section will grow in size and then eventually split into two new sections. There are numerous requirements that have to be met before a 'section split' is allowed. Thus each 'new' section is then responsible for half of the 256-Bit address range. As more and more complete *groups* of 8 vaults join the network, it continues to split and each section is therefore responsible for the curation of less and less data. An important thing to note is that the SAFE Network doesn't assign 256-Bit addresses based on proximity, in a given section two vaults could be very close together in 256-Bit XOR space but be located on different continents. This property helps to ensure the integrity of the network, by trying to ensure that vaults in a given section are not located close to each other it helps to increase the resilience of the network to attacks. If a significant number of vaults leave the network then 'sections' have the ability to join with other sections to ensure the stability of data is maintained.

Vaults can be characterised as having different 'Personas'. The most *basic* persona that a vault can have is that of the Data Manager. A Data Manager is responsible for the storage of chunks within a section. Their job is vital to the stability of the network. When data is stored on the network, it is actually 'duplicated' across multiple Data Managers. At all times the network aims to keep a minimum number of copies of a chunk of data, if a chunk goes missing (say a vault goes offline) this chunk is quickly duplicated to another Data Manager to ensure that data is stored redundantly. Hence within a given section, there will be several vaults storing identical chunks of data. Each having full knowledge of the chunks of data that the other Data Managers hold. The other persona a vault can take is that of the Client Manager. A Client Manager is responsible for storing the account data for clients. When you create an account on the SAFE Network, that data is stored like any other piece of data on the network. It has a given 256-Bit Address and contains the information like: how much Safecoin an account has, the number of chunks of data that has been uploaded, etc. As an account is a 256-Bit address it will fall within the domain of a particular section, the Client Managers in that section will then store the relevant data. As I will discuss in the section on Encryption, a vault does not know the IP address of the client that it is interacting with. The Client Manager thus doesn't know the IP address of the client it belongs to, it is just data and they cannot arbitrarily read the account data because it is encrypted.

2.3 Crust and Encryption

Crust is the secure routing layer used by the SAFE Network. It was designed and built by Maidsafe to provide the secure communications backbone of the SAFE Network. Crust allows for reliable peer to peer connections and provides encryption for all traffic. I won't go into too much detail but some important points to realise is that Crust doesn't have a standard port required to function, it is capable of randomising ports. Several Transmission Protocols can be used, falling back to UDP from TCP (for example) if required. Encryption at this level means that Data on the network is always encrypted, data is only decrypted client side and whenever it is not on a clients computer it is fully encrypted.

Encryption is a very important aspect of the SAFE Network. Whenever data is stored on the network, it is encrypted. As mentioned above the only time data is unencrypted is when it is on a client. Data on the network exists as discrete 1Mb chunks, each with its own 256-Bit Address. When a file is uploaded to the network, it undergoes a process known as self-encryption. Self-Encryption is a pioneering technique developed by Maidsafe and is used to encrypt data. What happens is that when your file is broken down into 1Mb chunks, each chunk is encrypted with the hash of one of the other chunks. What happens then is a DataMap is constructed, this DataMap then contains the addresses of each of these individual chunks of data so that they can be retrieved. As data is stored on the network in this manner, you have a number of options on how to access it. You can choose to have data "unencrypted" or what Maidsafe calls "Plain", what this means is that any user that knows the address of the data (and the type-tag) can retrieve and read the data. The special thing about this is that the data is still fully encrypted on the network through self-encryption, a vault owner cannot decipher what the chunk of data holds. When anyone goes to access this data though, it is reassembled and you can read it. The two other types of encryption supported are Symmetric and Asymmetric. Having these options means that you can build applications in quite a flexible manner. A user can freely share the key to data and this opens up the possibility for interesting designs.

A system is also in place to protect a users identity as they connect to the network. This aspect of the SAFE Network is very important to my project. When a client connects to the network, they do so through the use of a *Proxy Node*. A Proxy Node is a vault that is used to liaise between a client and the network at large. When a user connects, the Proxy Node of course knows the the IP address of that client. Beyond the Proxy and deeper into the network all the vaults know is the XOR Address of the account being used (including other relevant data and public encryption keys). Hence by using a Proxy Node, the activity of the client is well hidden from the rest of the network. A given vault cannot detect that the data being retrieved is going to someone in a particular country etc. This means that clients can anonymously read and store data to the network without people being able to monitor the contents of that data. I will touch upon this later on when I discuss my project, this anonymity is very important.

2.4 Quorom and the Datachain

As the network acts as an autonomous entity, there has to be some method for a given vault to reach consensus with other vaults. This problem is what Cryptocurrencies aim to solve through processes such as mining. Mining is essentially the network reaching consensus upon what has happened (in this case, financial transactions). In the case of Bitcoin, every time a block is mined, it is cryptographically linked to the block that came before it. As this *Blockchain* grows in size, the consensus on past transactions grows and grows. For Bitcoin and similar cryptocurrencies, to be able to undo a transaction/block you would need to have control of over %50 of the networks hash power. The debate on how easy it is to do that is a hotly debated topic that is outside the scope of this paper. The SAFE Network needs a similar mechanism on how to reach consensus. Analogous to a Blockchain, the SAFE Network has a 'Datachain'. This Datachain is used to help insure the integrity of the network and can be used to help rebuild the network incase of a catastrophic failure. For any action on the network to be valid, whether this be the storing of data or a vault joining a section, there has to be a corresponding 'group signature'. This group signature is stored in the Datachain that all vaults in a section has. In order for an action to be valid, a section has to reach a 'quorum'. For a network where the minimum section size is eight, a quorum would be five out of the eight vaults. This means that in a given section, several vaults could be acting as 'bad parties' but network integrity wouldn't be lost. XOR Distance also comes into play in this process. The closer two sections are in 256-Bit XOR Address Space the more they know about the data the other section is storing. They will have access to the portion of the Datachain that is used by that section. This way, a given section can help to verify that a neighbour is acting as a good party in the network and that data being stored there has not been tampered with. The further away in 256-Bit Address Space two sections are then the less they know about each other. This means that as the number of sections increases, the influence a given a section has over the network decreases. Eventually resulting in no section in the network having an overview of the entire

network.

A protection mechanism exists in the retrieving of data to account for the case when a vault tampers with data after it has been recored in the Datachain. When a client requests a given piece of data, a single vault is chosen to return that chunk of data corresponding to a 256-Bit address. Alongside the data that is returned, a minimum number of acknowledgements from other vaults in the section must be returned too. This way, a client can then verify the data they receive against the acknowledgements from the other vaults in order to ensure that the data is valid.

The development of the Datachain is still very active, at the time of writing I have tried my best to summarise the current proposals. Things are subject to change as Maidsafe runs simulations and exams how things operate.

2.5 Node Age and Churn

A crucial part of the integrity of the Datachain is node ageing. In order for a vault to *vote* on network activity (this is the signatures that form the group signature) it has to have proved itself a reliable party. A vault cannot just join the network and start voting in network decisions. When a new vault announces itself to the network, it is issued with the Proof of Resource that we discussed earlier. If it passes the proof of resource then as long as the assigned section reaches a quorum on the new vault joining, then it joins that section. This node is very 'young' in the eyes of the network and as such is not trusted. It is not allowed to vote in group actions and is responsible only for the storage and transmit of data. A very interesting aspect of the SAFE Network is the concept of *churn*. Churn is used to constantly 'rotate' vaults round different sections on the network. This means that in a given time frame, a vault will not be responsible for the same 256-Bit address range. This important feature helps to ensure that it is very difficult to track down where data is stored in order to erase it or corrupt it. During churn, young vaults with a lower node age will be chosen more frequently than older vaults. The vault is assigned to a new section, to which it must give another proof of resource. If the new section reaches quorum then the vault joins that new section and its node age is incremented. Thus, trust must be earned by acting as a good party in the network over time. Only when a node reaches a certain node age does it become an *elder*. An elder is a node which has a high node age, meaning it has been up and running for a while and has proven itself to be a reliable party. When a node is an elder, it gains the voting writes that eventually lead to the construction and maintenance of the Datachain. Vaults that are not elders have no voting writes and essentially just do what they are told by the elders. If a vault acts out of order then its node age can be decremented or eliminated entirely. Trust must be earned.

Node ageing and churn are hence essential security features of the network and make it very difficult for an attacker to have any choice in the section of the network they wish to attack.

2.6 Data

Data stored on the SAFE Network can take one of two forms. It can either be *Immutable Data* or *Mutable Data*. A Mutable Data Structure (often abbreviated MD) is a key value storage mechanism that allows for the storage of 1000 entries at a maximum size of 1Mb. A MD has a 256-Bit address to specify its location. An Immutable Data Structure only stores a single 'value', its address on the network is derived from the hash of binary data it contains. An Immutable Data structure can itself only be 1Mb in size, but through the use of a Data-Map this limit can be subverted. I will talk more in depth about the Data-Map when we discuss my project, as its properties are very important to my application. As their names imply, Mutable Data can be freely changed and updated whereas Immutable Data cannot. Its clear to see that if you change the contents of Immutable Data then it will no longer correspond to the address at which it is stored. As mentioned previously, it is this property of

Immutable Data that eliminates duplication on the network. If a user uploads the same file as another user they are simply presented with another *key* to access that data. If a user 'deletes' the data it will remain on the network as another user still maintains the key to access it.

Bibliography

[1] Maidsafe. (2018). Secure access for everyone, [Online]. Available: https://safenetwork.org/ (visited on 01/16/2018).

[2] ——, (2018). The world's first autonomous data network, [Online]. Available: https://maidsafe.net/ (visited on 01/16/2018).

[3] N. Lambert. (). Autonomous data networks and why the world needs them, [Online]. Available: https://blog.maidsafe.net/2017/10/07/autonomous-data-networks-and-why-the-world-needs-them/ (visited on 01/16/2018).

[4] Wikimedia-CH. (2018), [Online]. Available: http://www.openzim.org/wiki/OpenZIM (visited on 01/16/2018).

[5] United States Holocaust Memorial Museum. (2018). Book burning, [Online]. Available: https://www.ushmm.org/wlc/en/article.php?ModuleId=10005852.

[6] C. Harlan. (Apr. 2013). In north korea, the state-run news agency is the weapon of choice, [Online]. Available: https://www.washingtonpost.com/world/asia_pacific/in-north-korea-the-news-agency-is-the-weapon-of-choice/2013/04/28/88f3003e-aff2-11e2-bbf2-a6f9e9d79e19_story.html?utm_term=.8c0c3036efac (visited on 01/16/2018).

[7] BBC. (Apr. 2017). Turkish authorities block wikipedia without giving reason, [Online]. Available: http://www.bbc.com/news/world-europe-39754909 (visited on 01/16/2018).