ELECTROENCEPHALOGRAM CHAOTIC CRYPTOGRAPHIC HASH FUNCTION

Adewale Temiloluwa Obadina
AO19206

BENG COMPUTER SYSTEMS ENGINEERING

CE301 INDIVIDUAL CAPSTONE PROJECT CHALLENGE

SUPERVISOR – DR. MICHAEL BARROS
Second Assessor – Dr. Jichun Li

Acknowledgements

The author would like to thank Dr Michael Barros for the supervision provided throughout the lifespan of this project and report.

Another acknowledgement is to be made to thank Henri Begleiter at the Neurodynamics Laboratory at the State University of New York Health Centre at Brooklyn for provision of the electroencephalogram data that was used in the development and testing of this project.

Abstract

The goal of this project is to provide a system in which users can produce a secure hash from their Electroencephalogram (EEG) data via a chaotic cryptographic hash function. This project attempts to provide a solution to the rise of cyber-crime. Two factor is now the standard for authentication, typically based on “what you know” and “what you have” factors of authentication (e.g., passwords and access to an email account). This solution is based instead on “what you know” and “what you are” factors. Theoretically this should be a more secure approach as replication of biometric information is much more difficult. A possible application could allow for users to create a hash from the EEG signals of a thought, which then gets stored in a server database in place of a typical text password hash. The user would authenticate their identity by reproducing the thought, in turn reproducing the same hash and authorizing them access. This program operates by performing a bitwise XOR calculation of the binary conversion of a chaotic tent map’s output with the binary conversion of the EEG data. The result of which is permuted and then compressed into a 256-bit fixed length hexadecimal message digest.

Table of Contents

I. List of Symbols

Electroencephalogram – referred to as “EEG”

Brain Computer Interface systems – referred to as “BCI(s)”, “BCI technology”

Sixth Generation of Cellular Networks – 6G

Bitwise Exclusive OR operation - XOR

II. Introduction

Passwords have been the default authentication method since before computers were invented. They are suitable in most applications, as when proper precautions are taken when creating them, they can prove to be secure enough to be practically unbreakable. However, the problems with passwords are becoming more apparent. What constitutes a “strong” password evolves with the current processing power available, as many more brute force attempts can be computed per second. This as well as the fact that malicious parties will become more experienced at attacking systems results in the complexity of a suitable password having to increase. Now, most services check the strength of a password before it is saved to the database, usually requiring a minimum of 8 characters, containing a mix of lowercase and uppercase letters, numbers and possibly symbols. This complexity makes attacking them much harder, but also makes remembering them much more difficult as well. With the explosion of online services in the past twenty years, an average person can expect have at 100 individual online accounts[1] they need access to. Ideally each of these accounts should have a completely unique and complex password to ensure proper security. In practice however, this can prove to be unreasonable without an application to keep track of them such as a password manager (which come with their own risks being single point of failure). This hefty requirement contributes to why users tend to have such weak, vulnerable passwords. The success of humanity can be partly contributed to our ability to recognize patterns as “Pattern recognition skills sit at the helm of our basic cognitive architecture.”[2]. Research claims it’s this ability that aided our ancestors in socialization and survival. In this circumstance it is at our detriment because when you need to remember multiple passwords, users are prone to having their passwords contain patterns. This is evident with the most common passwords for multiple years being some deviation of “123456”, “qwerty” or “password”[3][4]. Each year the general population becomes more security conscious with headline after headline of malicious data breaches and leaks. The fact that these passwords are still so incredibly prevalent indicates that either people are indulging in optimism bias (believing that they could never be the victim of an attack) or consciously use these passwords because they struggle to remember 100+ unique passwords.

The main motivation for this project is to provide an alternative, more secure solution to the problem of modern user authentication. Most modern authentication systems are based on two-factor authentication, typically being “What you know” (a password) and “What you have” (access to a specified email account or phone number). The third type of factor of authentication is “What you are”. This factor is based around biometric information and the principle that it is much more difficult to replicate, however not equally difficult for all methods of biometrics. For example, DNA and fingerprints are more distinctive than signatures or voice[5]. Although previously being used in certain niche applications, the “What you are” factors of authentication has seen a large push into the mainstream. This is with the introduction of Apple’s Touch ID and Face ID technology, as well as other companies like Samsung, PayPal and NatWest implementing their own biometric security measures into their products. However, the author believes that there is room for the application of biometric authentication to further expand. If the task of remembering a combination of mixed-case letters, numbers and symbols were reduced to remembering a single phrase per account, it would drastically reduce the over-reliance people have on using very basic passwords or the same password for multiple different accounts. Brain Computer Interfaces (BCI’s) have the potential to make this a reality. The project goals can be defined as:

- Produce a method of using chaos theory to create an effective hash from electroencephalogram data to provide authentication required for transmission over a high throughput wireless network.
  - Evaluate this solution's security effectiveness
  - Evaluate the overall effect Brain Computer Interface technology may have on society

As aforementioned, humanity has become reliant on technology for society to properly function. This can be substantiated with the Office for National Statistics (ONS) reporting that in 2015, 11% of adults in the UK had never used the internet before[6]. However, a 2020 report claims that now 92% of UK adults were recent internet users[7].

The value of the data stored is proportional to the use of the technology. Therefore, as we increase the ways in which technology can be applied to our lives, we also increase the incentive for malicious third parties to attack and steal the data which we hold precious. Cybercrime has been steadily increasing in prevalence, but research shows that the COVID-19 pandemic may have exacerbated the problems. Research shows that there was a 43% increase in cybercrime from before the pandemic started to after it[8].Considering the large influx of users working from home due to government regulations, this has potentially opened a large amount of security vulnerabilities. The increase of cyber-crime stresses the important of being more security conscious, which is why the abundance of weak passwords is such a problem.

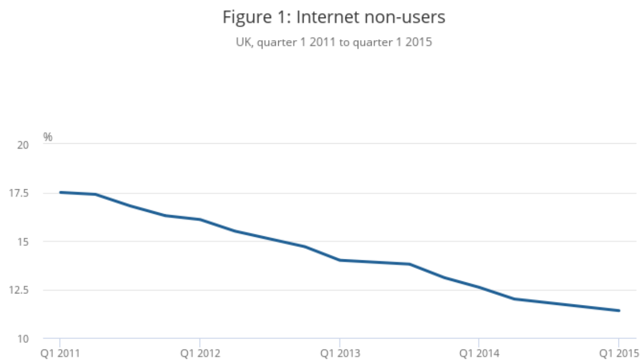


Figure 1: Statistics of Internet non-users in the UK
Source: Office for National Statistics[6]

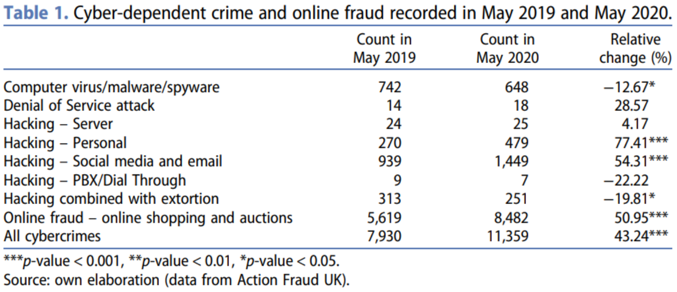


Table 1. Cyber-dependent crime and online fraud recorded in May 2019 and May 2020.

	Count in May 2019	Count in May 2020	Relative change (%)
Computer virus/malware/spyware	742	648	−12.67*
Denial of Service attack	14	18	28.57
Hacking – Server	24	25	4.17
Hacking – Personal	270	479	77.41***
Hacking – Social media and email	939	1,449	54.31***
Hacking – PBX/Dial Through	9	7	−22.22
Hacking combined with extortion	313	251	−19.81*
Online fraud – online shopping and auctions	5,619	8,482	50.95***
All cybercrimes	7,930	11,359	43.24***

***p-value < 0.001, **p-value < 0.01, *p-value < 0.05.
Source: own elaboration (data from Action Fraud UK).

Figure 2: Cybercrime comparison from before and after the start of the pandemic
Source: Office for National Statistics[8]

To demonstrate a theoretical application of this solution in society, consider a scenario where technological development progressed towards the widespread adoption of wireless BCI implants and the sixth generation of cellular networks (6G). The reason 6G would be a pre-requisite is because of the small size of a BCI implant, a higher bandwidth frequency would be required for quick data transmission, which 6G promises to provide with frequencies up to 3THz[9]. In this proposal, a user has access to a personal BCI and wants to log into an online service. This solution could allow the user to authenticate themselves without them having to type their password. Instead, they could think of a particular thought or do a certain action and the EEG signals produced by the thought/action would act as the "password". It is widely agreed that storing a plaintext "password" in a database is bad security practice, so the proposed solution would create a fixed hash from the signals created which gets stored in the server database. This means that the next time the user wants to log onto the service, they only need to replicate the thought or action that produced the hash that is stored in the database. This allows for the user to remember their passwords much easier if they only must remember a single thought or action as opposed to a potentially unintuitive string of characters. This would also provide improved security because the biometric data will be specific to that user, therefore more difficult to forge. Another, slight, benefit is that this approach dampens the effectiveness of keylogging malware if a user never actually types their password.

Apart from the security benefits, this solution can be classified as assistive technology for people with disabilities because it could provide the user more agency and independence when accessing potentially sensitive information online. In this scenario where BCI's become more prevalent, people with chronic disabilities may be given an option for implanting a BCI for the independent supervision of their medical conditions, similarly to how diabetic patients can monitor their blood/glucose levels. Consider another user with a personal BCI who suffers with motor difficulties such that typing on a keyboard is difficult. This solution would allow them to authenticate themselves without having to share that password to a third party to log on for them or say the password loud for voice recognition, potentially saving them from a malicious act from a third party.

The vision being portrayed is too idealistic without considering the potential drawbacks of such a society. The hurdle of BCI's being accepted into the mainstream consciousness is a difficult one as there are concerns that must be deliberated for the preservation of safety for the general public. The physical concerns about BCI's mostly pertain to the risks of implanting the technology into a user's skull. This can present problems such as skin erosion and abnormal bone growth[10]. But since the alternative of non-invasive technology seems to be a safer options, the ethical concerns are more pressing. If not regulated properly problems such as thought policing and malicious marking practices can very well be a real threat. These issues are discussed in more detail in the second portion of the literature review; "The Benefits and Potential Dangers Surrounding Brain Computer Interfaces". The author believes that the issues surrounding BCI technology are of the utmost important and need to be resolved before its widespread commercialisation. If that can be done, then the benefits that BCI's can provide to society could be exceptional.

III. Prerequisite Knowledge

Due to the nature of this project being interdisciplinary between computer security and neuroscience, the following paragraphs are dedicated to presenting and explaining the core topics that are encompassed within it.

❖ Brain Computer Interfaces (BCI) and Electroencephalograms (EEG)

A Brain Computer Interface (a.k.a. Brain Machine Interface) is a system which enables real-time, direct communication between the brain's electrical activity and an external computer device. The scope of what is considered a BCI is limited to systems that only interact with the signals created by the body's central nervous system[11]. There is a misconception that BCI's can be used to "read minds" of (non-consenting) users. However, a more apt description would be that a BCI acts as a "mental peripheral", allowing a user to send commands to a computer system without having to use a typical peripherals such as a mouse and keyboard.

There are three categories BCI's fall into; invasive, partially invasive, and non-invasive BCI's. Invasive BCI's require that a device be surgically implanted into the grey matter tissue of the user's skull whereas partially invasive BCI's rest outside of the brain tissue in the skull. Non-invasive BCI's only require an electrode cap to be worn by the user.

There are different methods in which BCI's can be applied to obtain brain signals. The scope of this paper will only focus on Electroencephalograms because they can be collected from the scalp as opposed to implants on the cortical surface. Although much simpler to apply than implants, this comes at a cost to spatial resolution. The flesh, bone, and hair between the brain's electrical signals and the BCI electrodes dampen the precision, with invasive BCI's having the potential to record the activity of single neurons[12]. The scope of this report will only focus on non-invasive technology. This is because the convenience of only having to wear a cap as opposed to neurosurgical implants mean gathering research participants is much easier, consequentially there is more research available for it. The second reason is that a very high level of spatial resolution is not required to reliably gauge the intent of a user from features extracted from the signals recorded. Furthermore, users face risks with invasive hardware, such as; infection, skin erosion and disrupted cellular milieu[13].

The application of BCI's at their inception were restricted to the "rehabilitation and medical care for patients to restore social interaction or movement capabilities[14]". The success of which inspired research into where else they could be utilized. Since then, BCI's have seen uses in the prevention, diagnosis, and monitoring of health conditions such as dementia, brain tumours and epilepsy. They are also used in the remote control of motorized wheelchairs[15] and prosthetics[16]. Although mainly used in medicine, BMI's show potential for use in many other sectors such as marketing, education, games, and entertainment.

❖ Cryptographic Hash Functions and Chaos Theory

Cryptography is the study of secure communication techniques in the potential presence of malicious third parties. The field has been explored for centuries and is based around the CIA triad, data Confidentiality, Integrity, and Availability. Cryptography is now more important than ever because of our modern reliance on computer systems. As a result of this, the information we keep on computers become increasingly valuable, meaning that attacks to steal personal information have more incentive and therefore are continually rising. There are many techniques that have been developed to reduce our vulnerability to these attacks, such as symmetric encryption, asymmetric encryption, and hash functions. The scope of this project will only consist of hash functions.

Hashing is a type of one-way encryption, meaning you cannot obtain the plaintext (input) from the ciphertext (output). Hash functions convert data of any variable length into a seemingly random, fixed length sequence of bits called a hash (or message digest). Generally speaking, which is represented in a hexadecimal format. This process creates a situation in which trying to get the original plaintext from the hash is very difficult, ideally impossible.

One of the key features of hash functions is that plaintext *x* will always produce ciphertext *y*; $H(x) = y$. However, if a single bit of plaintext *x* gets changed such that *x* becomes $\bar{x}$, the hash produced will be different so that $H(x) \neq H(\bar{x})$. When small change in plaintext results in the large change of the ciphertext, this is denoted as the avalanche effect. This property is core to hashing algorithms as if two different plaintexts produce the same hash, then the hash cannot properly be used to identify what plaintext the hash is referring to, therefore the hash must depend on every bit of the plaintext. For example, if plaintext *a = "Good morning"* and plaintext *b = "Good night"*, but they both produce the same hash such that *H(a) = H(b) = "A1B2C3"*. How can an instance of the hash be used to confirm whether the original plaintext was a "Good morning" or "Good night" message? A hash function is said to be functional if:

- The input can be of any length
- The output is of a fixed length
- Its computationally simple to compute the hash for any given message.
- Its infeasible to create a plaintext that produced a specified hash.
- Its infeasible to modify a plaintext without modifying the hash.
- It is infeasible to find two different plaintexts with the same hash.

An example of how hash functions are used today is in the storage of sensitive data, such as passwords in databases. Storing plaintext passwords is a huge security risk and horrible practice as if an attacker gets access to the database, every user's password will be visible to read and steal. Instead, a better solution is to hash the user's password and store the hash. When the user enters their password, the system will hash what they entered. If the hash of the password they entered matches the hash in the database, it can be safely assumed that the characters are identical, and they are granted access. This is much safer as if someone were to break into the system and gained access to the login database, they will only see the hashes stored. Assuming the plaintext password sufficiently secure, breaking the hash could take tens of thousands of years to break.

Chaos theory was born from the study of non-linear dynamic systems and presents the idea that situations we understand to be random and unpredictable, are bound to underlying principles such as patterns, repetition, and self-organisation. The butterfly effect concept is a theoretical example of chaos theory. It states that a small difference in one state of a deterministic nonlinear system, consequentially results in a much larger difference in a later state. Essentially a small change in a systems input can have a domino-like effect and result in a very large change in its output. This property is identical to the aforementioned avalanche effect of a well-founded hash function algorithm.

IV. Literature Review

As stated in the introduction, the motivation of this project is to further research into the proposition that the information collected from BCI technology can be applied to the field of cryptography as a means of biometric authentication. Therefore, the scope of the literature being reviewed is restricted to these three themes; [1] cryptography utilising chaos theory, [2] the benefits and potential dangers surrounding Brain Computer Interfaces, and [3] the suitability of brainwaves in biometric cryptography. After conducting research on the aforementioned themes, the conclusion has been reached that cryptographic solutions that incorporate chaos theory are a mainstay in cryptography techniques that are used in today's landscape. The author of this paper will attempt to contribute to with their own version of such. The benefits of BCI's are exhaustive and well documented, however this may not be true for the risks. Proper education to the public of what the technology can and cannot do will be required to combat the exaggerated claims some media articles make. When (more so than if) the technology arrives to a commercial market, government regulations enforcing ethical design frameworks are imperative to secure the safety of the populace. Finally, there is evidence to support the hypothesis that brain data can be used as a method of biometric authentication. In fact, in comparison to other biometric methods, brain waves may be the best solution for some of the problems that other methods cannot solve.

❖ Cryptography utilizing Chaos Theory

The concept of using the qualities encapsulated within chaotic systems in cryptography was first investigated in 1989[17]. After the recent discovery that *"well defined physical systems have the ability to exhibit "chaotic" behaviour"*, Robert Mathews presented the hypothesis that this technique could be applied to cryptography. With success he derived a formula in which he was able to hand encrypt and decrypt plaintext with just the use of a calculator. The method of which he used to do this was to select two numbers as parameters for his function, labelled β and x_0. The number of iterations of the function was equal to the length of the plaintext being encrypted. Each iteration of the function would produce a decimal between 0 and 1, which he took the two least significant digits of each decimal, calculated the modulo 25 and shifted the characters of the plaintext according to the result. For example, if the first output of the function was the value 0.812980077, the two smallest digits, 77, would be reduced to modulo 25 to obtain 2. The character would be shifter up by two such that *C* becomes *E*. In the context of the time, this approach was ground-breaking as Matthews states that there was "*difficulty of securely distributing the sequences of random numbers*", in which he demonstrates how previous methodologies would exhibit predictable behaviour, so a method of generating a reproducible sequence of highly sensitive "random" numbers was truly revolutionary.

However, the biggest issue with this methodology is that it only contains substitution, not permutation. This flaw means that it is very susceptible to frequency analysis, where the most frequent occurrences of the substituted letters can be compared to the most frequent occurrences in the English language, and therefore can be broken accordingly. Another issue is that Matthews bases his brute force security analysis on a machine consisting of 1000 T-800 Inmos Transputers, each having a computing power of 2MFlops[17][18], in total, providing around 2GFlops of computing power. Today, a premium commercial processor would be the Ryzen ThreadRipper 3990X, which has a reported 32-bit floating point operation per second of 13TFlops[19], approximately 6500 times faster than the machine Matthews refers to. Meaning the cipher, he claims would have taken 1000 years to break, can be broken in 54 days by a single processor today.

Since Matthews research, there has been much more development in the topic of chaotic cryptography, but because this project is focused on hash function specifically, the literature analysed reflects that. For example, a 2009 paper[20] proposes a more sophisticated interpretation of Matthews work. They present a hash function solution based on a one-dimensional nonlinear dynamical system, which is the same type of system Matthews used. This system is known as a tent map, where similar to Matthews, it the output is dependent on the two parameters in formula, as well as the authors claim, *"for certain parameter values, this system can display highly complex behaviour and even chaotic phenomena"*. Their method consists of "encoding" the plaintext into multiples of 1024 bits (it is assumed, but not explicitly stated, that the plaintext is first converted to binary), a single "1" bit and as many "0" bits are appended until the length is of a multiple of 1024. The binary is then partitioned into *n* blocks. The authors describe the following process as "encoding", but from their diagram below, the following procedure can be interpreted; The number of outputs of the tent map is (again assumed, not explicit) to match the number of plaintext blocks. A bitwise XOR operation is performed on the tent map output (K_n) with the binary plaintext (M_n) until all of the plaintext blocks have been operated on. The resulting data is then compressed into 128 hexadecimal string.

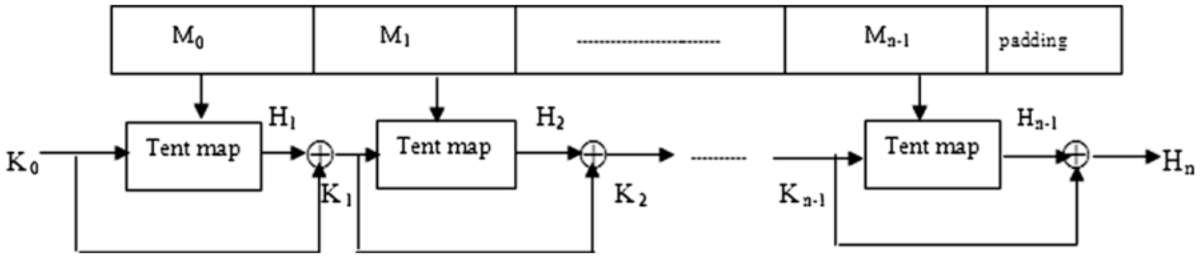


Fig. 5. Detail construction of CBHF.

Figure 3: Diagram outlining algorithm logic
Source: Chaos-based hash function (CBHF) for cryptographic applications[20]

The paper analyses their work and in it, they claim to have properly achieved the avalanche effect from a change in a single character. This is an important feature that is absent from Matthews' work as it proves that the whole ciphertext is dependent on each part of the plaintext. This means that the properties of confusion and diffusion are present in their solution. Which they also analysed and state that *"the ideal diffusion effect should be that any substantial changes in initial conditions lead to 50% changing probability of each bit"*. After performing a test where a single random bit is flipped, and a new hash generated, the original hash is compared with the new hash and the difference in changed bits is counted. The test is repeated 1024 times and they report *"the mean changed bit number and changed percent is 63.84 and 49.88%, respectively, which are very close to the ideal value 64 and 50%"*. The solution is then claimed to be sufficiently resistant to birthday and man-in-the-middle attacks.

Critiques of this publication begin with the lack of detail present in their methodology. With such a brief explanation of how their solution operated, a lot of assumptions needed to be made on the reader's end. With no pseudocode to account for this it is unclear how reliable their procedure, and consequently, their results are. Secondly, only qualitative analysis is provided for birthday and man-in-the-middle attacks. Because of the lack of code description, these claims cannot be peer-tested. The assumption that all 128-bit hashes are secure is fallacious, as the MD5 algorithm can produce 128-bit message digests[21] but is widely known to be vulnerable to collision attacks. Therefore, this assumption could lead to an oversight in tested security in that area, as the level of is dependent on the input data as well as the implementation of the hash function. This project will aim to improve upon this works accomplishment via being more transparent about its operation as well as using a 256-bit hash by default to improve upon its resilience.

❖ The Benefits and Potential Dangers Surrounding Brain Computer Interfaces

With any scientific pursuit, the good that the endeavour can produce must be evident. In the 50 years of research since first being introduced by Jacques Vidal in 1973[22], there is overwhelming evidence to support the benefits that brain computer interfaces can provide. A 2012 paper published by Jerry Shih, Dean Krusienski, and Jonathan Wolpaw discusses how BCI technology is used and where their future may lie. The authors define a BCI as *"a computer-based system that acquires brain signals, analyses them, and translates them into commands that are relayed to an output device to carry out a desired action.*[11]*"*. They specify what does and does not classify as a BCI by stating that only signals produced by the central nervous system are collected by a BCI, which excludes technology such as voice-activated and muscle activated systems. They also make the claim that electroencephalograms alone are not BCI's since they only record the brain signals but does not use the signals to generate an output to interact with the users physical or digital environment. Another important misconception that is debunked is that BCI's are "mind reading" devices. *"Brain-computer interfaces do not read minds in the sense of extracting information from unsuspecting or unwilling users but enable users to act on the world by using brain signals rather than muscles*[11]*"*. This distinction is important because there seems to be exaggerated misunderstandings of the actual capabilities of the technology, as will be explained in a later paragraph. However, it is important to know that this is beyond the scope of what BCI's are capable at the moment, but considering the rate at which technology develops, how long until this feat is no longer an exaggeration.

Shih, Krusienski and Wolpaw document a brief overview of the development of BCI capabilities. By 1980, EEG activity was used to control the vertical movements of a rocket travelling in a television screen[23]. 1988 showed that the P300 event-related potential could allow users to spell words on a computer. In 2006 a tetraplegic patient was implanted with a microelectrode array, the signals were used to successfully open a simulated e-mail, operate a television, open, and close a prosthetic hand and perform simple actions with a robotic arm.
There are plenty of other examples of researchers successfully using BCI's to influence the physical world via prosthesis[24], orthosis[25][26], electric wheelchairs[27][28][29] and more. The same can be said for the digital world with users being able to control cursors in 1, 2[30][31][32] and 3[33][34] dimensions.

With regards to critiques of this paper, the authors make the claim; *"In all hands, no matter the recording method, the signal type, or the signal-processing algorithm, BCI reliability for all but the simplest applications remain poor"*. It is unclear what the authors define as "simple applications" but there is evidence that support very high accuracy rates for different BCI tasks. Although this study was published in 2012, there are papers from as early as 2000 who report respectable accuracy rates. Such as this paper[35] which reports accuracies averaging 87.7%, 74% and 77.5%. As well as this paper[36] which claims; *"The grand average mean accuracy was 87.4% for all patients and sessions. All patients were able to achieve at least one session with a maximum accuracy above 96%"*, which brings Shih, Krusienski and Wolpaw's claim some contention. One very important caveat to notice is that the majority of these studies use very small sample sizes of a uniform set of subjects, all subjects tend to have very similar characteristics. The first paper referenced [35] only had a sample size of 3 male, right-handed healthy patients, with ages between 17-26. Whereas the second has a size of 3 men and 2 women, an average age of 60 and all had suffered an ischemic stroke in the territory of the sylvian artery. A 2020 paper[37] reports very high accuracy results of *"96.83% and 91.86%*

for EEG literate and EEG illiterate respectively. However, the sample size was also very specific, consisting of five men and five women, an average age of 23 ± 1.56, all right handed with normal eyesight and no history of brain related diseases or brain damage. This makes it difficult to assume the successful data can be extrapolated to have the same results in a much larger populous. Therefore, before a concrete conclusion can be reached, more research needs to be conducted with a wider and more diverse sample size. More development of the technology may be required to make BCI's more accessible to more researchers in order to conduct more large scale experiments.

As with any system, as well as their benefits, BCI's pose risks to their users. A 2019 paper published by F. Gilbert, C. Pham, Jnm Viaña & W. Gillam analysed 4064 articles from a FACTIVA database on BCI's, from their first mention in the media in 1993 to the end of 2017. They reported: *"76.91% of articles portrayed BCI positively, including 25.27% that were overly positive… In contrast, 1.6% of articles had a negative tone and only 2.7% of articles flag issues explicitly related to ethical concerns surrounding BCI technology* [38]*"*. These statistics show a clear bias from the media to overstate the benefits of the technology and downplay the potential risks that are involved, especially with overzealous claims in the likes of; *"…we could live healthier and longer lives, and see huge progress in easily overcoming cultural and linguistic barriers. It is likely that a human with no need for sleep or food may emerge on earth* [39]*"*. This is a troubling realisation considering some of these article's border on propaganda, without stating that the potential risks are real threats, and so the public must be made aware. This portion of the literature review will divide risks into two categories; the physical risks they pose, as well as the societal and ethical risks.

When discussing the physical risks, it is clear that non-invasive BCI's are marginally safer than invasive ones, as the removable nature of them prevents users from being exposed to long-term health complications. This is true currently, as the consensus is that 5G technology is safe for humans; *"A European Commission expert committee concluded that current knowledge about how electromagnetic fields (EMF) interacts with the human body could be used to set exposure limits for the whole frequency range up to 300 GHz*[40]*"*. However, considering that 6G will see deployment in the coming years, the frequency range of up to 3THz will question whether BCI's that use 6G frequencies will be safe for long, or even short term use. Supposed solutions are put in place to make the transmission of data safe for humans, the health implications of implanted technology still remain. All surgical procedures carry risk of complications. Eran Klein outlines these risks, defining an exhaustive list of risks in the following areas: *Perioperative Electrodes, Electrode Biocompatibility, Electrode Durability, Neuroplasticity, Electrode Obsolescence, Wired, and Battery Power Systems, Wireless or Inductive Power Transmission Systems, Wired and Wireless Data Processing and Data Processing Obsolescence.* [41]. Although it can be argued that as the field develops more, the chance and damage of the risks will decrease or be mitigated, even the simplest of procedures still have chance complications today. Invasive BCI's are reported to have superior recordings to non-invasive; *"…the fact that the recording electrodes that are placed under the dura leads to higher spatial resolution than do EEG* [42]*"*. But since there is (albeit unrepresentative) evidence of high accuracy results via non-invasive, BCI's, the author of this paper questions whether the health risks are worth the higher precision readings, and if so, in which circumstances.

With regards to the societal and ethical risks, a March 2022 article reports that almost 60% (4.62 billion) of the world's population actively engages with social media platforms[43]. Which is a 115% increase from the 2.07 billion in 2015[44]. The advancements in computational speed, machine learning sophistication and improved economics of data storage has resulted in companies being able to offer attractive and personalised user experiences. However, it is these same advancements that have given birth to the nefarious and extremely profitable market of user data. Srdjan Lesaja and Xavier-Lewis Palmer present concerns about the current state of the digital space, and where its future may lie if precautions are not pre-emptively enforced before the commercial deployment of BCI technology. Lesaja and Palmer use the term "neuro-capitalism" to describe *"an economic subsystem surrounding the collection, commodification, use, and brokerage of any measure that serves as proxy for a neural state*[45] *"*. They begin their works by defining problems people currently face in the "digital space" which they define as the Internet, and all network-connected computers, devices, and storage. The way in which companies and corporations exploit their users are explored. This includes techniques such as drafting excessively long terms and conditions contracts, filled with vague and complicated language. Which is done to prey on how the onus of education regarding data safety is on the user, and how the majority of the population will not participate in their elucidation. The contracts obfuscate clauses which contain how a users' data will be stored, processed, and possibly transferred to third parties.

Companies claim that they give their users a choice with the argument that these services are purely optional, and any user can simply opt out of using them. However, Lesaja and Palmer combat this, indicating that in today's world, this can become practically impossible; *"Many large corporations that engage in the surveillance economy have worked to make their platforms so pervasive that to avoid using them takes sometimes overwhelming effort and would put people who make this choice at a significant disadvantage."* Suppose a privacy-conscious individual chooses not to engage in these services. This would mean forgoing critical ones like an email account. Considering a vast majority of the ways we interact with the world is predicated on owning one, such an individual would make the process of tasks like finding jobs, accommodation, and education much more difficult, if not, impossible. If the same were to be said for BCI's in the far future, people may be cornered into owning one, subsequently forfeiting a choice to not have one.

Machine learning is becoming more specific to target the individual as opposed to a population. This means that engaging with these services will give companies access to subconscious information that we ourselves may not be aware of, such as browsing and purchasing habits, political and social views, and other people we associate with. All for the purpose of tailoring experiences so we are more likely to continue to use these services, in which they can collect more data to train their machine learning algorithms, to show advertisements specifically targeted to users as individuals. This need to keep users in their ecosystems to further mine user's habits is evident in how online services, especially social media, are designed to manipulate the fundamentals of neuroscience to keep users perpetually involved, hence the field of neuro-marketing[46]. Although not recognised by official mental health governing bodies, studies have concluded the symptoms of overuse in social media include low work performance[47][48], less healthy social relationships[49][50], sleep problems[51][52], low life satisfaction[53][54], and feelings of jealousy, anxiety, and depression[55][56].

The point being illustrated is that some corporations will pursue revenue at seemingly any costs, using manipulation tactics to profit from their userbase, regardless of any physical or mental anguish they may experience. The significance of this concern will only exacerbate with the question; If these are the problems we face today, how many more issues will arise if the data containing how and what we think became a commodity? Lesaja and Palmer define a list of possible consequences of corporate BCI exploitation, including that of thought policing, othering, targeting of vulnerable groups (showing a recovering alcohol addict with beer advertisements, for example) and strengthening the wealthy, with more social, economic, and political influence over society. These sentiments are also echoed in other papers[57].

Three solutions to these issues are proposed by Lesaja and Palmer, which are education, ethical design, and regulations. *"An educated and sceptical public is the last, and also best defence against the dangers outlined above."* Considering the complicated nature of how BCI's work, it is of high importance that the public is well educated on exactly how the technology operates, as well as the potential benefits and risks clearly defined. This would allow for people to make a properly informed decision on whether they decide to participate in its use. One can make an argument that the reason the public needs to be informed is because placing trust in engineers and developers to make systems that are ethically correct is a dangerous investment. Especially considering the conflicting interests in the companies that fund the research may prefer otherwise. Large companies like Facebook[58] and Google[59] have expressed their interest in BCI field, but their ethical practices have been in question[60][61]. Therefore, it is difficult to determine how much faith the public can place in large companies to keep users' best interest in mind. Anyhow, with well-defined and strict regulations on how and what companies can do with such data, governments can prevent such companies from leveraging their power against the wellbeing of the public.

After evaluating the claims made in this work, the author of this paper is in firm agreement, as these are problems society are currently struggling with. Therefore, the advent of commercial BCI's will raise the limits to which the general public can be exploited. This paper will not attempt to provide a solution to these issues; however, the author acknowledges that they must be highlighted as serious problems that need to be solved in due time. Lest society moves into a dystopian future, where even one's own thoughts are not safe monitoring and abuse.

❖ Using Brainwaves as a Means of Biometric Authentication

The last theme to be covered is whether the electrical impulses in the brain can be used as a suitable means of authenticating one's identity. A 2020 paper published by Fares Yousefi, Hoshang Kolivand and Thar Baker proposes a signal acquisition system for user authentication and authorisation. This scope of this review will be focused on the how brainwaves are presented as an authentication method, and as such will not explore the specifics of the system the researchers present. The paper claims that *"...human biometric techniques are the most secure methods for authentication purposes that cover the problems of older types of authentication like passwords and pins*[62]*"*. This is supported by quoting eight advantages and six drawbacks of biometric systems from this paper[63].

The benefits are listed as: [1] Improved security, referring to biometrics as "*nearly impossible to hack unlike passwords*". [2] Physical traits such as fingerprints, face and retinas are consistently accurate. [3, 4] Biometric credentials are a part of you, making the need to memorize anything redundant, therefore convenient, and flexible. (Convenience and flexibility are defined as two separate points, but the provided definitions are essentially identical.) [5] *"Reports claim that the young generation trust biometric solutions more than other solutions."* [6] Biometrics are better scaling solutions that others, [7] Biometrics are time efficient to use. [8] Biometrics are cost effective for the company.

The drawbacks to biometrics are as follows: [1] Face, fingerprints, retina, and iris information are non-cancellable. Unlike a password, which can always be changed to another, these biometrics are fixed. Meaning once they are imitated, they will never be a secure method of identification. [2] Prone to false acceptance rates (FAR) and false reject rates (FRR). FAR's are much more of a concern than FRR's, since the biometrics in the database can be reconfigured to mitigate this, whereas FAR's will give a user incorrectly authorised access. [3] Some biometric systems can take longer to properly authenticate than is ideal. [4] Contact based biometrics like fingerprint scanners may be used by a large number of people. Especially since the COVID-19 pandemic, people must be more conscious of personal hygiene. [5] Unfortunately, some people may have damaged or lost body parts required to engage in these systems. Meaning alternatives must be applied in these circumstances to prevent disability discrimination. [6] The environment and usage of the systems can affect the measurements taken by it. The authors then compare the qualities of some biometric identifiers with those of brainwaves. The qualities being judged are based on this[5] paper and are defined as such;

Universality *Each person should have the characteristic.*

Distinctiveness *Any two persons should be sufficiently different in terms of the characteristic.*

Permanence *The characteristic should be sufficiently invariant (with respect to the matching criterion) over a period of time.*

Collectability *The characteristic can be measured quantitatively.*

Performance *Which refers to the achievable recognition accuracy and speed, the resources required to achieve the desired recognition accuracy and speed, as well as the operational and environmental factors that affect the accuracy and speed.*

Acceptability *Which indicates the extent to which people are willing to accept the use of a particular biometric identifier (characteristic) in their daily lives.*

Circumvention *Which reflects how easily the system can be fooled using fraudulent methods.*

Note that there is a discrepancy in the table name of the characteristics, as Yousefi, Kolivand and Baker (seemingly mistakenly) refer to “Acceptability” as “Accessibility”.

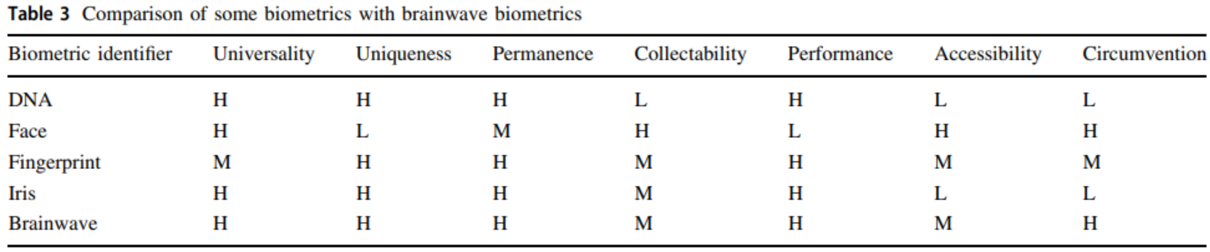


Table 3 Comparison of some biometrics with brainwave biometrics

Biometric identifier	Universality	Uniqueness	Permanence	Collectability	Performance	Accessibility	Circumvention
DNA	H	H	H	L	H	L	L
Face	H	L	M	H	L	H	H
Fingerprint	M	H	H	M	H	M	M
Iris	H	H	H	M	H	L	L
Brainwave	H	H	H	M	H	M	H

Figure 4: Comparison of the characteristics between common biometric methods
Source: SaS-BCI: a new strategy to predict image memorability and use mental imagery as a brain-based biometric authentication.[62]

From how the pros and cons of biometrics are presented, it is clear that brainwave biometrics share all of the benefits while omitting some of the disadvantages. For example, face, fingerprints retinas and other biometrics are non-cancellable, but this drawback does not apply to brainwaves, as the brainwaves reflect the mental state of the subject as well as what they are focusing on. For example, if a user was authenticating themselves with brainwaves containing the thought of a dog, and that data was stolen. The user could change the brainwaves used to authenticate to that of a cat. The point of some biometrics being unhygienic could apply to brainwaves, depending on the method of extraction. It does not apply to partially invasive/invasive BCI’s because they are implanted in the user’s skull. But may be true for non-invasive systems if users had to share devices like caps between them. Considering the risks associated with invasive, the author of this paper is partially biased to non-invasive BCI’s therefore this downside will still be held against brainwaves as an authentication method. Considering the abundance of research supporting how brain damaged patients can still engage with the system, the problem of physical disabilities preventing usage is a non-factor. As long as the subject is alive and conscious, they are able to engage with the system.

Although not a direct critique of Yousefi, Kolivand Bakers work, they referenced data claiming that one of the supposed advantages is how young people supposedly trust biometric solutions more than others. Looking at the source paper[5], there was no citation attached to this claim. Therefore, it is difficult to know what statistics the researchers were basing this conclusion on. Although, the concept that younger people are more willing to adapt to new technologies is not surprising, considering that older generations find it more difficult to overcome habitual responses[64].

A 2012 American survey[65] supports this sentiment. It was conducted on 134 young (between 18 and 30) participants. Although the gender split is a fair 60% female to 40% men, the paper notes that 82% of the participants were Caucasians with around the same level of education (76% graduated from US college) and earning the same yearly income (91% earned $25,000 or less). Whether the sentiments produced from this survey would be consistent of a larger demographic is unclear. One of the first questions asked is how much the participants knew about biometrics, to which 74% responded they had never heard of them before. The paper claims to have evidence to support the idea that young people are welcoming to biometric usage, however, upon perusing though the statistics, the participants remained mainly neutral when discussing using biometrics to access services like online shopping or a personal computer, with the exception being accessing their car. They seemed to be much more supportive of biometrics when in the context of Institutions like hospitals or bank accounts requiring them as authentication. When asked about how they felt about the reliability of biometrics compared to passwords, the results clearly shows that the consensus was that they were an improvement.

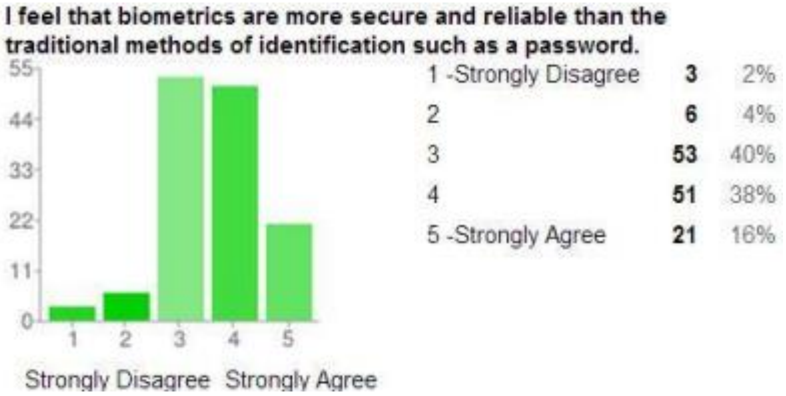


Figure 6: Representation of how young people feel about the reliability of biometrics as an authentication metod
Source: Young Adult Perception and Acceptance of Biometric Technology[65]

Although the author of this paper believes that this survey may not be credible enough to draw strong conclusions, considering how biometrics are much more integrated into society now than they were ten years ago, it is very likely that the public is in even more favour of their widespread use. This provides confidence in the idea that people are looking towards biometrics as a possible replacement for solutions like passwords as pins. If not a replacement, at least a primary authentication method with passwords and pins being a secondary, back-up method. Relating this back to this paper, it supports that brainwaves can be suitably used as biometrics, and could arguably be the best form of authentication, providing more development is done in the area.

V. Technical Documentation

The desired function of this project was to be able to create a system in which a secure message digest can be produced from a recorded instance of EEG recordings. A pseudocode representation of the code created is included in the corresponding sections of the documentation. The methodology is a modification of the counter block cipher mode of operation. This is to allow encryption to be processed in parallel, allowing for increased performance because the block encryption is not dependant on the previous operation being completed first. The algorithm logic is based on the following 8 step process:

1. Convert stream of input data into binary equivalent.
2. Divide input stream into "blocks" of 128 bits and store into a list.
3. Generate a list of chaotic tent map values, the same length as the number of elements in the binary array.
4. XOR each *nth* element of the binary array said *nth* element from tent map and store in a new list.
5. To compress list into one hash, XOR the first element of the list with a defined initial vector and then permute the result.
6. Move onto the next element of the list and XOR with the result of the previous operation, then permute the output.
7. Repeat step 6 until the list has been completely iterated through.
8. Convert the final output into a hexadecimal equivalent.

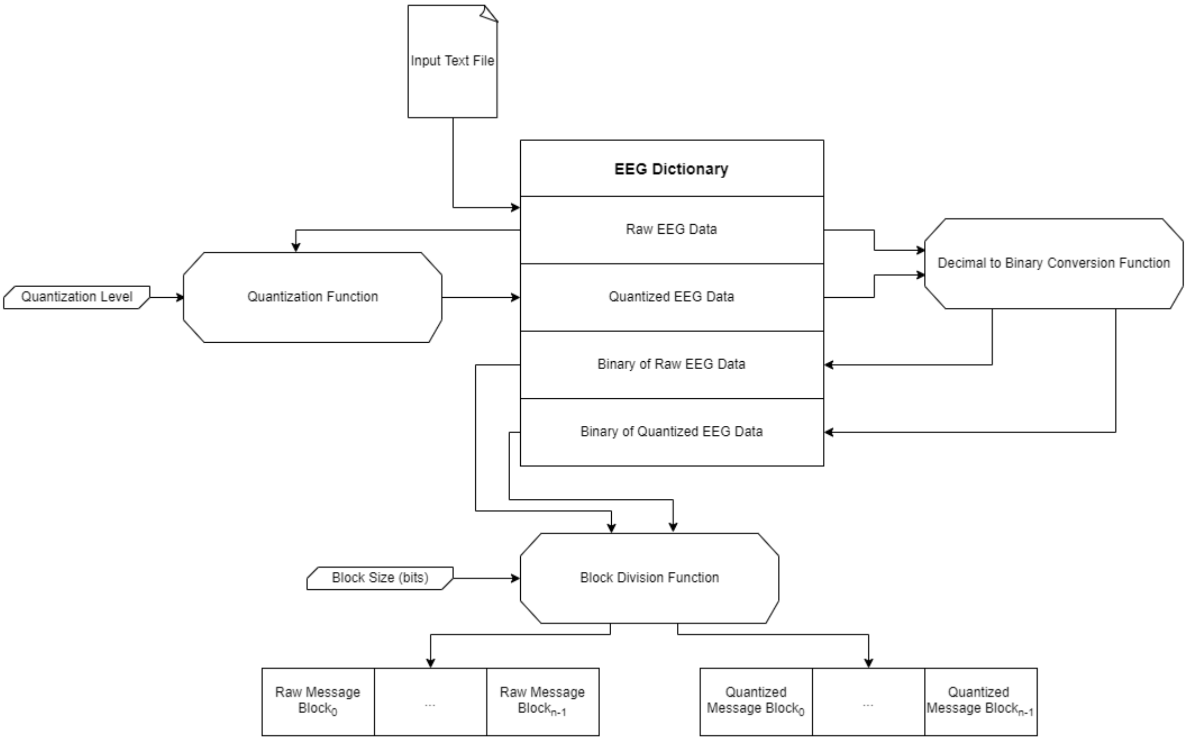


Figure 7: Diagram outlining parsing the text file, quantization, binary conversion, and block division
Source: The author of this paper

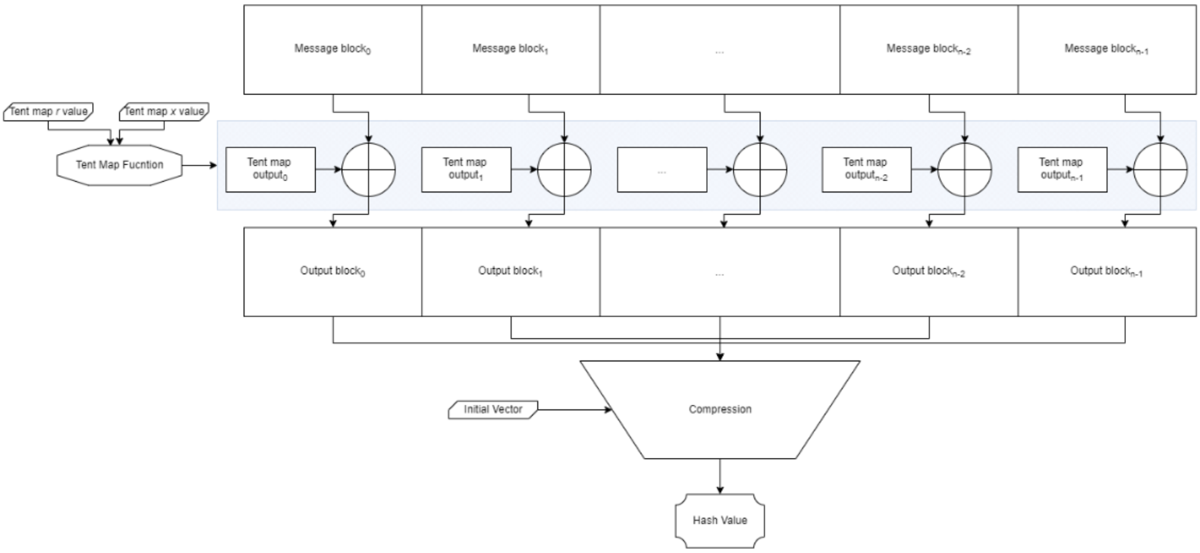


Figure 8: Diagram outlining the hashing process
Source: The author of this paper

The images above portray an overview of the program algorithm. The first diagram demonstrates a text file being parsed, and the “raw” EEG readings are stored into a list within a dictionary. The raw data is quantized to a specified quantization level and stored in a separate list. Both the raw and quantized EEG data are then converted into binary and stored into their respective lists. Each verson of the data is then divided into blocks of a specified bit length. The second diagram shows the hashing process once the binary blocks have been prepared. This hashing process is done twice in tandem, once for the raw EEG data that is being hashed, the second is for a quantized version of the EEG data. There are two reasons that the data is quantized. The first is that by quantizing (rounding) the data being processed, the system can save memory processing the hash, meaning that in a use case scenario, the response time of the system should be quicker. Quantizing the data reduced the bits needed to represent the data from 32 bits to 8. Given the computer used in development and testing had an octa-core CPU with 16GB of ram, the difference in processing times is negligible. However, in the scenario where this solution is used on an integrated circuit with a much more restricted set of computational resources available, the benefit of quantization would be much more apparent. The second reason that the data is quantized is to account for minor inconsistencies when attempting to reproduce a thought. Given the sensitive nature of the electrical activity in the brain, it can be assumed that purposefully attempting to recreate such activity can be difficult. This can be due to variables that are impractical for one to control, such as the unconscious activity of the automatic nervous system, meaning that reproduction of an instance of the brains activity are prone to small inconsistencies. The act of quantization attempts to account for this by reducing bits used to represent a signal, in turn reducing the degree to which two readings can be considered different. It is important to note that doing such comes at the increased risk for collisions, because there is a more limited set of variables to represent a limitless combination of electrical recordings. However, the practicality of quantization in real use case scenarios is too important to concede. Furthermore, a strong and well-founded algorithm for producing a hash can make the risks of collisions inconsequential.

The way in which the algorithm with be explained will follow the algorithm's order of operation. The first function to discuss is how the input data was converted into binary. At the beginning of the development cycle, the program was first developed with text as opposed to EEG data. This was because converting text to binary is a relatively simple operation. Therefore, other functions of the program could be created and tested sooner on the text based data, as opposed to the time it would have taken to search for the EEG data itself and implement around it. When the program was functional and tested with the text based data, that's when EEG data was implemented into it.

❖ Parsing the EEG Text File

The method chosen to store the data read from the file was a dictionary. The reason being it was the simplest method to store multiple arrays per EEG channel. The dictionary is defined with 64 keys, each representing an EEG channel. Each channel key has a value paired to it, which is a list of four lists. The lists are assigned as the first being where the raw EEG values from the file is read into. The second list is the binary conversion of each value in the raw list. The third list is the EEG data from the first list, after it has been quantized to a specified amount. The fourth is the binary equivalent of the quantized data. The terms "key" and "channel" will be used interchangeably. They both refer to the keys of the EEG dictionary, referring to the corresponding EEG channel in the brain.

The function that manages reading the file takes no arguments and is called from the main function of the program. The current working directory of the program is recorded and saved into a variable. This is because during testing, omitting the full file path when entering the name of the file to open caused unintended behavior. Therefore, the program will take the file path of the directory the program was launched from to save a user having to unintuitively enter the full file path. The program will assume that the directory it is launched from contains a folder with the EEG text files to operate properly. An infinite while loop is started, in it there are "try" and "except" statements. In the "try" statement, the program will prompt the user of the name of the target file to be used as an input. If the name of the target file is a match, the file is opened, and the loop is broken. In the case where the file name is not found, a message is alerted to the user, the loop repeats and they are prompted for the file name again. The EEG files are taken from an online database[66]. The EEG text files are formatted in such a way that there is one reading per line. The first word for each line is always a 0. The second is the name of the channel the reading is from, the third is an index of each reading per channel, and the final is the reading itself. There are a few lines of description at the start of the file which are ignored. Instead of manually deleting the irrelevant lines, the program will ignore them. Once the file is successfully opened, a for loop begins which iterates over each line of the text file. The first action in the for loop is to split the string of the contents of a line into a list of words, separated by whitespace character. If the third word of the list is not numeric, then the line is skipped. This is to ignore the description lines at the start of a file. Another for loop is started which iterates over each key in the EEG dictionary. When a key in the dictionary matches the second word in the line, then the last word of the line is appended to the first list in the dictionary. This populates the first (raw data) list for each key in the dictionary.

```
Function open_file (No  arguments):
        Get filepath of current working directory

        While true:
                try:
                        Output prompt to for user to enter name of target file
```

```
            Open target file
        except:
            Output that target file name does not exist

    For each line in file:
        Split string of line contents into list of words

        If the second from last word in list is not numeric:
            continue

        Else:
            For each channel in the EEG dictionary:
                If the second word matches the name of a channel:
                    If the third word is not "chan":
                        Append the last word of the line into
the first list of the EEG dictionary
    Close file
end
```

❖ Quantizing the EEG Data

Now that the raw EEG data has been read into the dictionary, the remaining lists of the quantized data and the binary conversions need to be filled too. The main function contains a short for loop, iterating over each key of the dictionary and calling a function to operate on them. This function will iterate though each element of the first list for each for each channel in the dictionary. Within this function, another function to quantize the data is called. Quantizing the data within the first list was done through a simple (almost) one line function. In which, the data to be quantized is the argument. The data is converted to a float and divided by the pre-defined quantization value. The built-in python function *round()* is called to round the result of the division. After rounding, the result is multiplied by the same quantization value and is returned. When the quantizing function is called, the result is appended to the third list of the dictionary. Below is an image of the raw and quantized EEG data for a test file appended together. Although the difference is too small to see, the quantized readings have been quantized to the nearest whole integer.

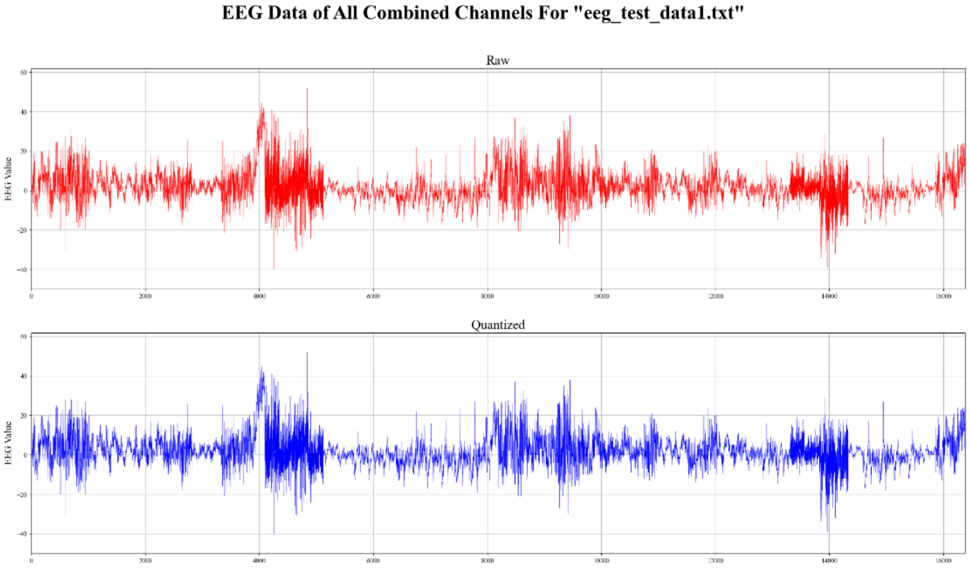


Figure 9: Graph of all readings of all 64 channels appended together for raw and quantized data
Source: The author of this paper

```
Function quantization(Argument one: raw data)
    Reference global variable "quantization value"
```

```
        Round the data to the nearest n, defined by quantization value
        Return result
end
```

❖ Converting the Data to Binary

Returning to the binary conversion function, the values in each list are split into two parts, the integer, and the decimal. There is a chain of three if-else statements to filter the types of numbers. If the integer part of the number was equal to 0, then only the decimal part is converted to binary, it is then prepended with as many 0's as is required to meet the length of 32 bits. If the integer is equal to 1 or above, the integer and decimal parts are both converted into binary and prepended with 0's so that they are of 16 bits each. If the integer part begins with a "-" sign (indicating it is a negative number), then both the integer and decimal parts are converted to decimal. The difference is that the twos-complement of the integer part is taken. This is to provide a distinction between the negative and positive binary numbers. The results of each are appended to the second list in the dictionary. The reason that the numbers are prepended with 0's to match 32 bits is to provide uniformity in the input when calculating the bitwise exclusive-or (XOR) operation of the binary numbers.

Another function is called to convert the quantized data into binary. It takes the quantized data as an argument and is a simpler version of the if-else chain previously mentioned as only the integer parts of the numbers are considered. The results of this call are stored in the fourth list in the EEG dictionary.

The binary values need to be prepared for further processing. So, in the main function, a for loop iterates through the keys in the EEG dictionary. Within this loop there are two more for loops which iterate though the data in the second and fourth list, the binary conversions of the raw and quantized data respectively. These loops will append all of the binary data together into one large string. When that is complete, a function is called to divide the stream into blocks, each with a length of 1024 bits.

```
Function rounded_data_to_bin(Argument one: data):
        If data is greater than 0:
                Convert data to binary string
                Strip string of "0b" binary prefix
                Pad the string with 0's on the left until the length is 8 digits
                Return result

        Else if data is less than 0:
                Convert data to binary string and remove "-" sign
                Strip string of "0b" binary prefix
                Pad the string with a single 1 and 0's on the left until the
length is 8 digits
                Return result

        Else if data is equal to 0:
                Return a string of eight 0's
end

Function convert_to_binary (Argument one: channel name)

        For each piece of raw data in channel name
                Split the integer and decimal parts into separate variables
```

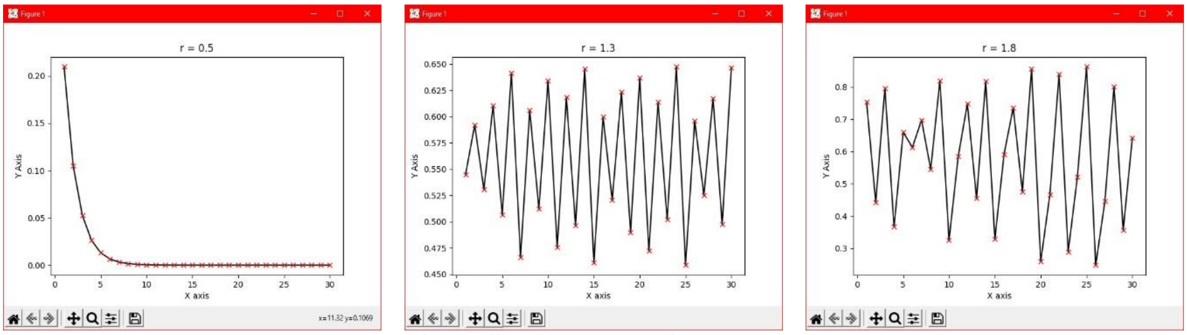
```
             If the integer part is negative
                    Convert to binary and take twos complement of the integer
                    Left pad the binary with 0's until the length is 16 bits

                    Convert decimal part to binary
                    Left pad the binary with 0's until the length is 16 bits

                    Combine the two parts and append result to second list in
EEG dictionary


             Else if the integer part is positive
                    Convert integer part to binary
                    Left pad the binary with 0's until the length is 16 bits

                    Convert decimal part to binary
                    Left pad the binary with 0's until the length is 16 bits

                    Combine the two parts and append result to second list in
EEG dictionary


             Else if the integer part is equal to 0
                    Convert decimal part to binary
                    Left pad  the binary with 0's until the length is 32 bits
                    Append result to second list in EEG dictionary

             Quantize the raw data

             Append quantized data to third list in EEG dictionary
             Call rounded_data_to_bin on quantized data and store result in
fourth list in EEG dictionary
end
```

❖ Generating a Chaotic Tent Map

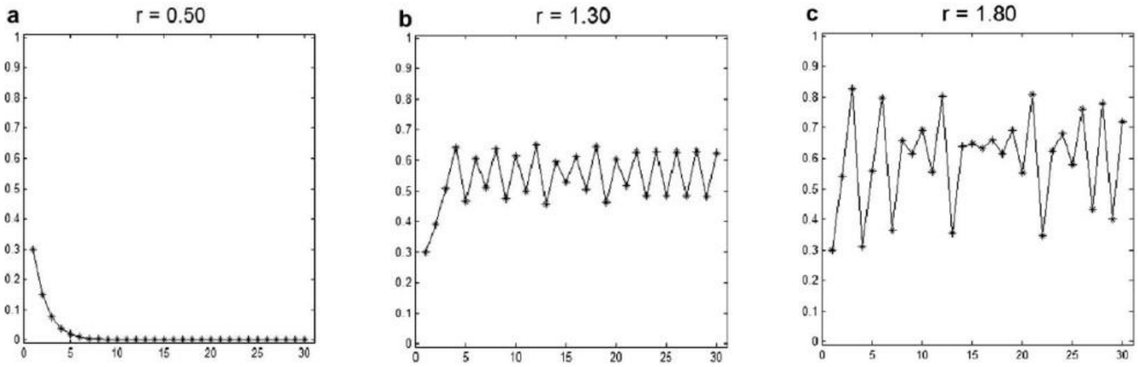
The next function in the process is the generation of a tent map. For this, inspiration was drawn from the 2009 paper[20] that was covered in the “Cryptography utilizing Chaos Theory” portion of the literature review. In which, the authors describe the formula of a tent map as:

$$T(x) = \begin{cases} rx, & 0 \leq x < 0.5, \\ r(1-x), & 0.5 \leq x < 1. \end{cases}$$

This formula takes in two parameters, *r* and *x*. The value for *x* is between the range of 0 and 1, while the values for *r* lie between 0 and 2. The *x* value acts as a sort of coefficient for the formula, while the behavior of the tent map is split into three distinct behaviors dependent on the value of *r*. *“When r ∈ [0, 1], the calculation results come to the same value after several iterations without any chaotic behavior. When r ∈ [1, √2], the system appears periodicity, and when r ∈ [√2, 2], it becomes a chaotic system with periodicity disappeared.”* (Quote adapted from[20]). The graphs shown below is a visual representation of this behavior. The author of this paper produced the top three graphs. The bottom three are from the paper where the quote was taken from. The graphs are clearly similar, proving that the implementation function of the tent map is operating as intended. The *x* value used in the CBHF paper was not explicit, which can explain the differences in the chaotic graphs on the right.



Figures 10-13: Graphical representation of the different behaviors of a tent map.
Source: The author of this paper



Figures 14-16: Graphical representation of the different behaviors of a tent map.
Source: Chaos-based hash function (CBHF) for cryptographic applications" for comparison.[20]

The implementation of this formula to python code is the authors' original solution, The function takes three arguments; *x*, *r* and size. The *x* parameter can be any decimal between 0 and 1. Because the chaotic behaviour is desired, the *r* parameter is kept between the square root of 2 and 2. It is encouraged that the *x* and *r* parameters be set to multiple digits after the decimal point to improve their robustness, therefore making them harder to be retrieved in the case of an attack on the algorithm. The "size" parameter of the function is equal to the number of blocks once the binary strings are appended together and partitioned into blocks. The function will output a series of chaotic decimal numbers between 0 and 1. The decimal part is converted into binary and stored in a separate list.

```
Function generate_tent_map(Argument one: Tent map X value, Argument two: Tent
map R value, Argument three: Size of message list to match)
        Create empty list for tent output

        For the size of the list to match:
                If X values is less than 0.5
                        X equals R * X

                Else if X is greater than 0.5:
                        X equals R * (1 - X)

                Convert result to binary and append result to tent output list

        Return tent output list
end
```

❖ Obfuscating the EEG Data

As shown in the diagram, a bitwise exclusive-or operation is performed with the nth block of the EEG binary data and the nth output from the tent map function. This is done via a simple function which takes the tent map output and the binary EEG as arguments. The result of which is appended to a separate list. Meaning now there are two lists which contain the obfuscated EEG data, one for the raw data and another for the quantized. At this stage, it should be difficult to retrieve the EEG binary plaintext without the *x* and *r* parameters that were used to create the tent map that it was XOR'd with.

```
Function xor_bits(Argument one: Binary tent map output, Argument two: EEG Binary
Block)
        Convert both argument to integers
        Bitwise XOR them together
        Convert the result back to binary

        Return result
end
```

❖ Compression into a Fixed Length Hash

The final step of the process is to compress the resulting list from the previous operation into a fixed length hash. This is completed by a function which is based on the Davies-Meyer single-block-length compression function. The function takes two arguments, the resulting list as well as a string to identify whether the list passed is for the raw or quantized datasets. The compression function works by firstly generating a second tent map with different *x* and *r* parameters to the one used earlier. This is set to the length of the resulting list. A while loop is set to iterate through each block in the resulting list. An if-else statement is used to filter the first entry in the list from the rest, as they follow two different operations. The first element of the list is XOR'd with a predefined initial vector, the result of which is permuted. The python functions from the "hashlib" library are used to store the message digest. This library provides a multitude of hashing algorithms, but the ones used in this program are the Secure Hashing Algorithms; SHA2 and SHA3. SHA3 is used by default but the user can opt for SHA2 when prompted when starting the program. Two variables from this library are created, they are updated with the first result of the XOR operation. For every element after the first , the result from the previous loop iteration (n_{-1}) is XOR'd with the current element (*n*). The result of which is then XOR'd again with the aforementioned tent map specific to the compression function. After which is permuted again, and the result is updated to the hash variables. This process is repeated until the end of the list is reached, where what is left is a single string of a fixed length hash. The result of which is converted into a 256 bit hexadecimal and presented to the user.

```
Function permutation(Argument one: binary string to permute)
        Split the string into four equal parts; s1, s2,  s3, s4

        Swap s1 with s3
        Swap s2 with s4
        Swap s2 with s3

        Join the parts together

        Return result
```

```
end

Function compression(Argument one: list of results from XOR operation, Argumnet
two: Flag string to identify whether list is raw or quantized data)
      Generate a tent map with different X and R variables to the one
previously created
      Reference global vraible inital_vector
      Set counter variable to  0

      While counter variable is less than the length of the argument list:

            If counter is equal to 0:
                  XOR the inital vector with the first block of the list


                  Call permutation function on result of XOR

                  If flag identifies raw data:
                        Update raw SHA variable
                  Else if flag identifies quantized data:
                        Update quantized SHA vairable

                  Increment counter

            Else:

                  XOR current block with previous result of loop

                  XOR result with corresponding tent map output specific to
this function

                  Call permutation function on result of XOR

                  If flag identifies raw data:
                        Update raw SHA variable
                  Else if flag identifies quantized data:
                        Update quantized SHA vairable

                  Increment counter
end
```

In terms of technical achievements of this project, all of the work undertaken is of the authors original solutions to the problems with two exceptions. The first is a solution for taking the twos complement of a binary number, which is an adaptation from that provided on an online Q&A website[67]. Secondly, the construction for the compression function is inspired by the Davies-Meyer solution[68], however the implementation is that of the authors.

VI. Results and Performance Analysis

The hashing algorithm developed in this project was relatively successful. The program is able take a text file containing 64 channels of 256 EEG readings each and output two message digests with a fixed length of 256-bit, one for the raw data and one for the quantized data. The program takes an average of 0.07 seconds to complete. This proves that the algorithm satisfies the quality of the hash function being simple to compute. It should be noted that testing took place on an octa-core desktop with 16GB of ram. Ideally the solution would have been simulated to check its performance in a scenario closer to its target use case, unfortunately time did not allow for such. Multiplying the processing time by 16 to account for higher machine specifications will still yield a time of 1.12 seconds, which the author still considers a reasonable time for a user to wait for a hash to be generated. This only considers the time it takes for the hash to be generated from the EEG data, it does not include the time required to collect the data, send the data hash to an authentication system, and wait for the system to confirm access. On the other hand, the algorithm developed will produce a hash for both the raw and quantized data, but in a use case, only the quantized data may be required. If such is the case, then the code can be refactored to only focus on producing the quantized hash, which can save possibly more time. An analysis was performed to evaluate the performance of the proposed solution. This encompasses assessing the programs levels of confusion and diffusion, distribution of hash values, resistance to collision attacks and its rate distortion for the quantized data.

❖ Confusion and Diffusion

Confusion and diffusion are two important properties of hashing functions. Confusion means that each bit of the message digest should depend on several parts of the key used to generate the hash. In the context of this project, the key can be considered to be the *x* and *r* parameters that are required to define the tent map function. Diffusion is defined as when a single bit of the plaintext is changed, then around half of the ciphertext bits should change as well. This relationship works inversely, so a change of a single bit of ciphertext should result in half of the plaintext bits being changed. A function that contains a good level of diffusion can also be said to contain the avalanche effect. Confusion is achieved by substitution in an algorithm, whereas diffusion by permutation. The reason these qualities are important is that they obscure the relationship between a cryptographic systems' key, plaintext and ciphertext, increasing the algorithms robustness by making it difficult to locate patterns.

The Hamming distance between two strings of equal length is defined as the number of positions in which the characters in that position differ. For example, 0110 and 0111 will have a Hamming distance of 1, whereas 0110 and 0001 will have a distance of 3. This method of comparing strings is how the confusion and diffusion will be measured. As described in the technical documentation, after the data was converted into binary, it was joined into a single large string and divided into blocks for a bitwise XOR operation to be performed on each block. The size of each block was defined at 1024 bits. An experiment was conducted in which a random single bit of the EEG data string was flipped. Subsequently a new hash was generated and the Hamming distance between the original hash and the newly generated one was calculated. This operation was repeated 10,000 times and the mean average distance was obtained. The rationale behind this is that the length of the hash produced is 64 hexadecimal digits, and according to the definition of diffusion, the Hamming distance should be at least 32. This program was repeated for 10 different variations of the size of the blocks the string was divided into, ranging from 32 bits to 16,384. This is to investigate whether the size of the blocks affect the confusion and diffusion properties of the hash function.

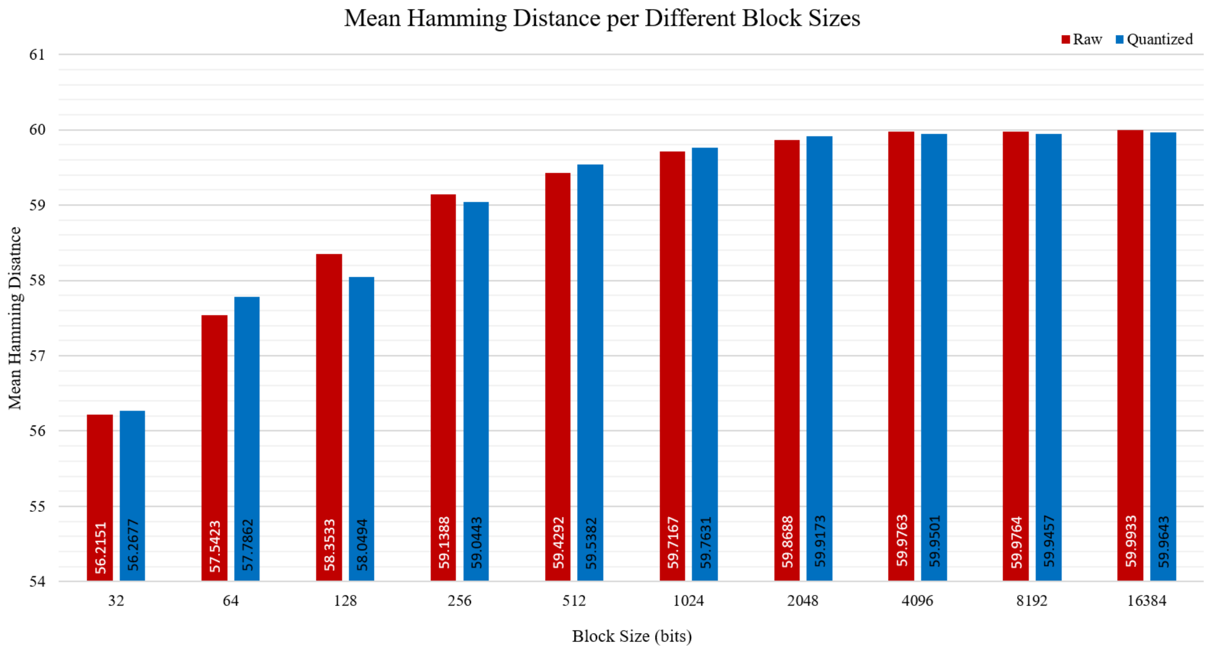


Figure 17: Mean Hamming Distance Bar Chart

The results of the analysis are presented in the graph above. The first fact to note is that the mean Hamming distance for all variations of the block size is above 56. This is well above the requirement of a distance of 32. Meaning that the results present a reliable level of confusion and diffusion. Another observation is that the block size does affect the Hamming distance to a degree. There is a sharp difference in the distance between block sizes of 32 and 64 bits, with the difference between each increment of blocks reducing until block sizes of 4096 bits, where the difference somewhat flatlines. This then suggests that the optimal block size to use would be 4096 bits, although this could only be true in the case of the length of the data set that was used in development and testing. It is possible that the optimal block size is variable, depending on the size of the data being processed.

It was also apparent that the block sizes also dramatically influenced the time it took to complete the program. Because the process of producing a hash was repeated 10,000, it exacerbated the processing time quite severely. This is apparent with block sizes of 8192 and 16,384 taking 147 seconds to finish, in stark contrast to 32 bit blocks taking 17 minutes and 45 seconds. Considering that a block size of 32 bits will require 512 times the number of operations that a block size of 16,384 will need, this dramatic comparison is understandable.

❖ Distribution of Hash Values

Uniform distribution of hash values is an important quality of a hash algorithm as it directly correlates to the security of the function. This is because if the values of the message digest are evenly spread, it makes it more difficult to recognise patterns within the ciphertext. The method used to analyse this quality is to generate a hash from a set of EEG data. A series of hashes will be generated based on how many random bits are flipped within the string of binary data that produced the original hash. This includes a single bit, 10 bits, 100 bits, 1000 bits and 10,000 bits being changed. The resulting hashes are then mapped onto a graph to analyse the distribution of the hash values. The results of which are shown below.

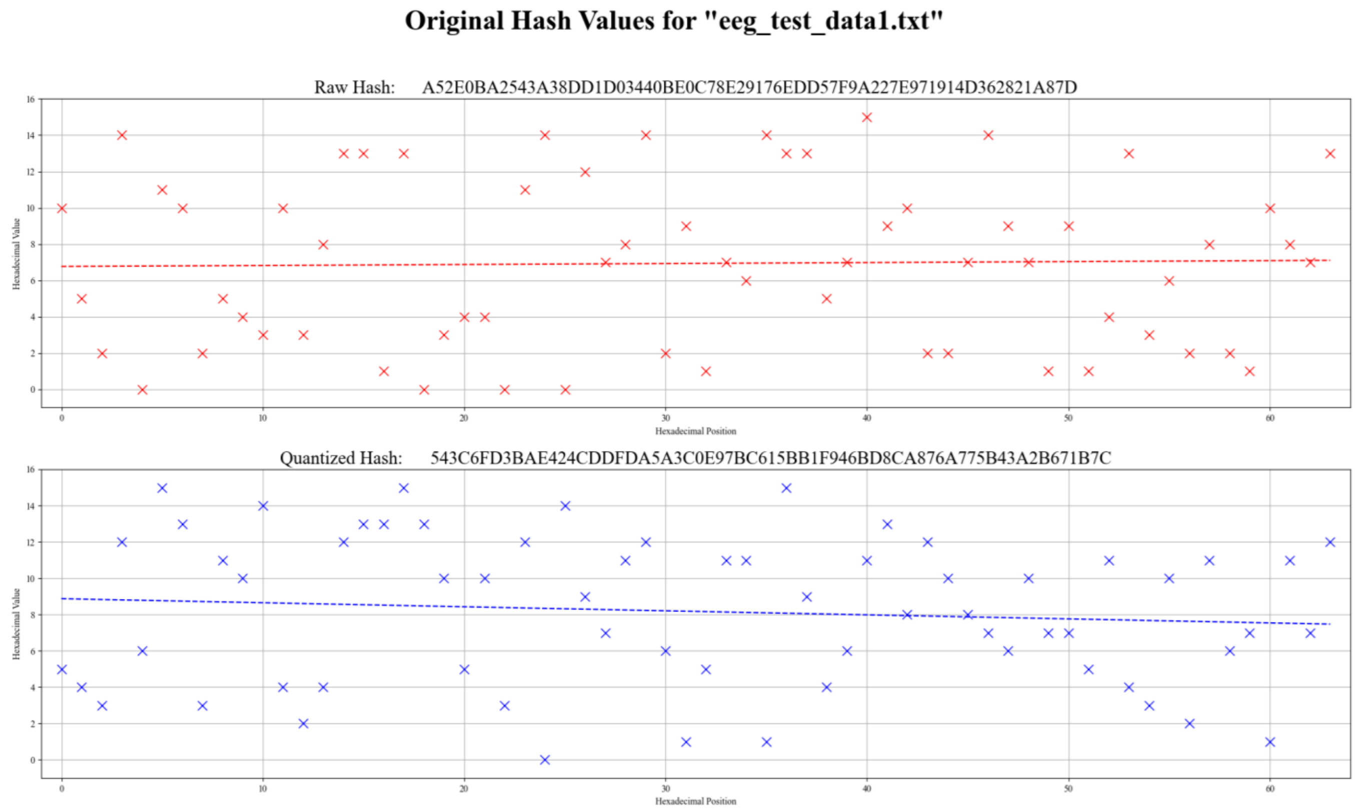


Figure 18: Hash Distribution of an EEG Data Set

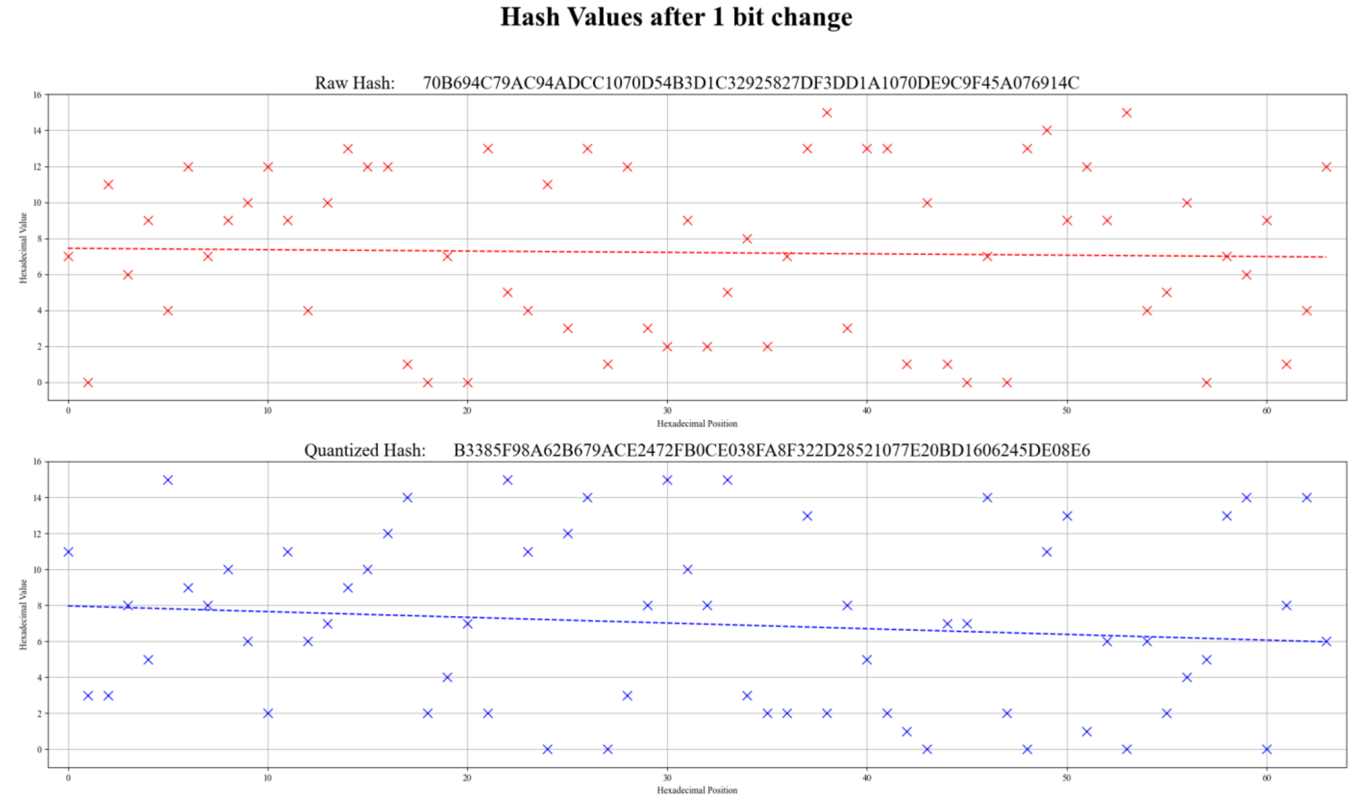


Figure 19: Hash Produced after 1 random bit change

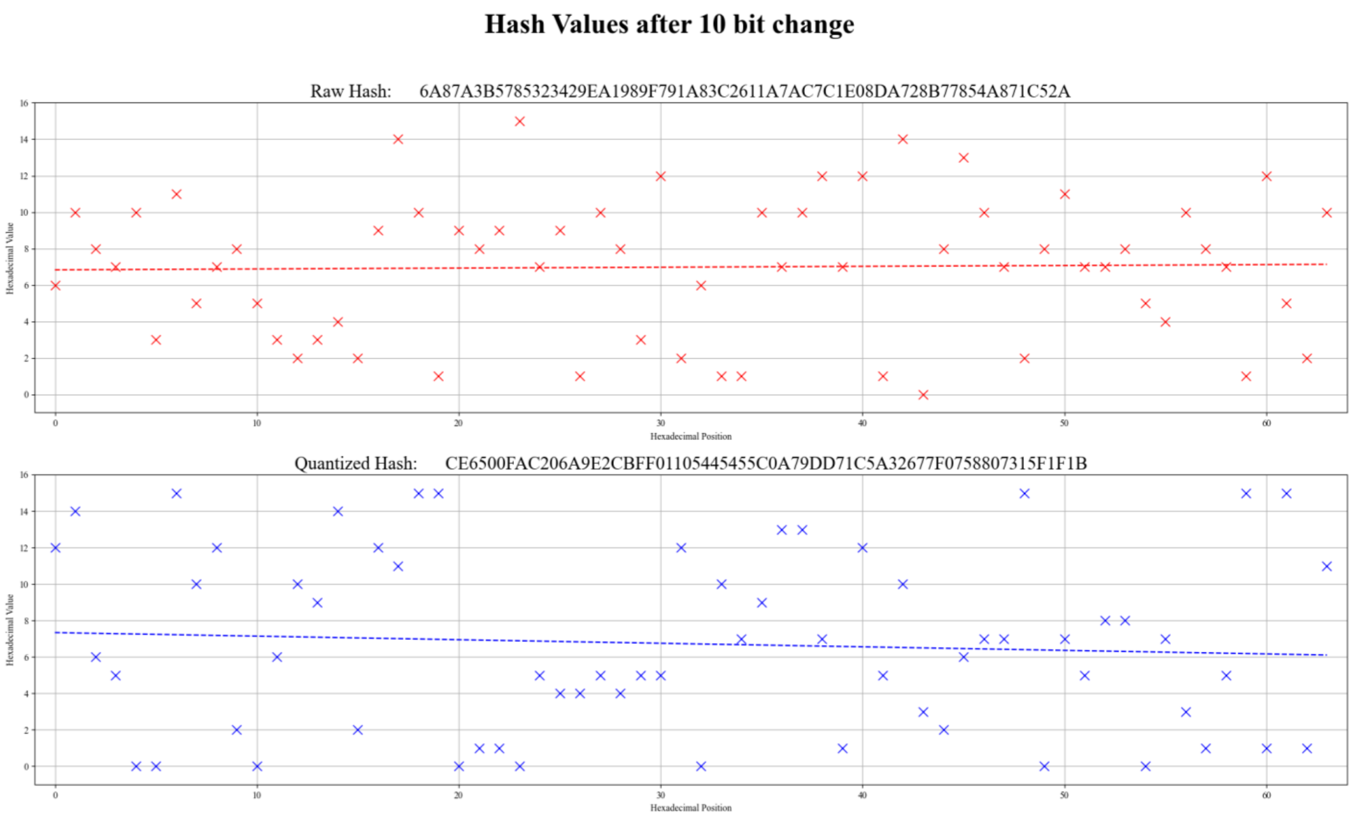


Figure 20: Hash Produced after 10 random bit changes

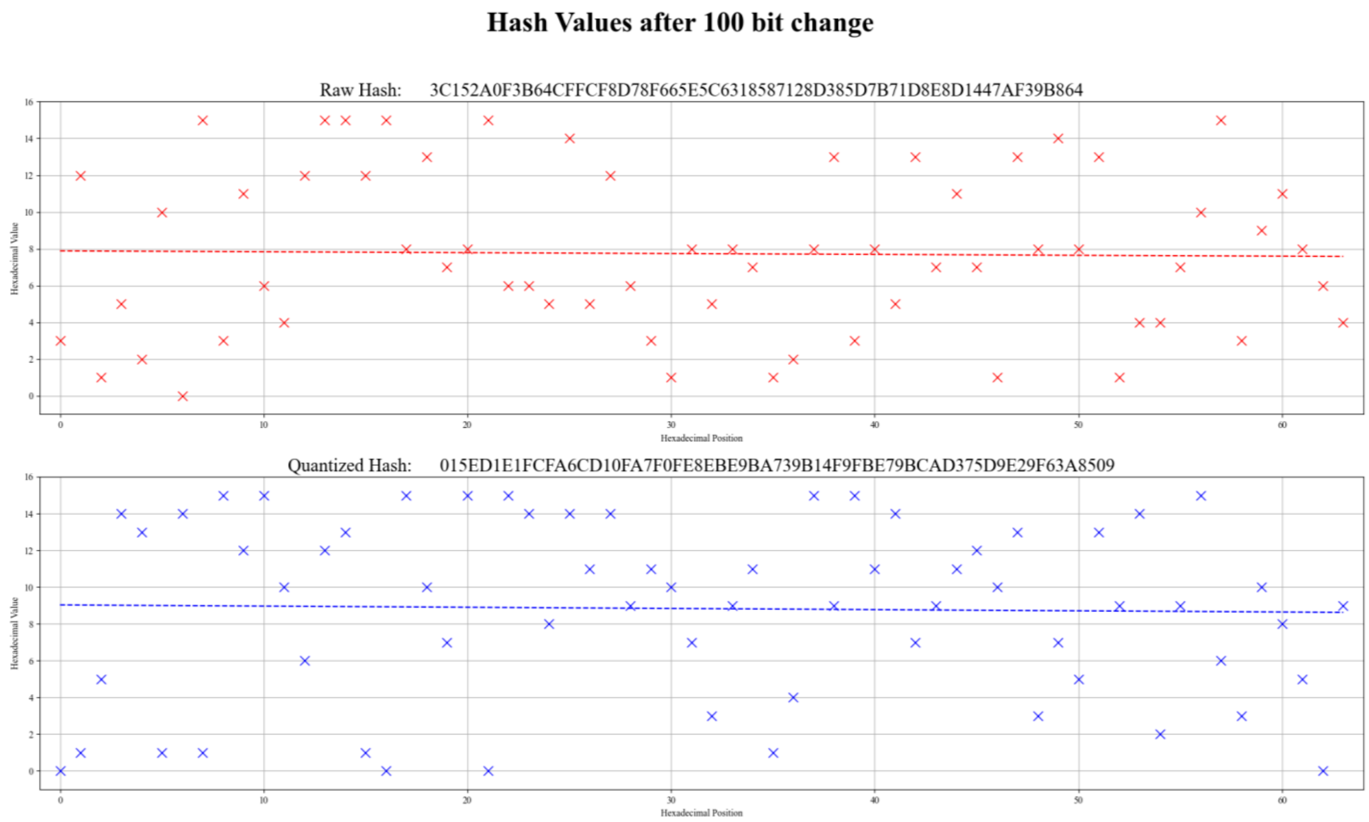


Figure 21: Hash Produced after 100 random bit changes

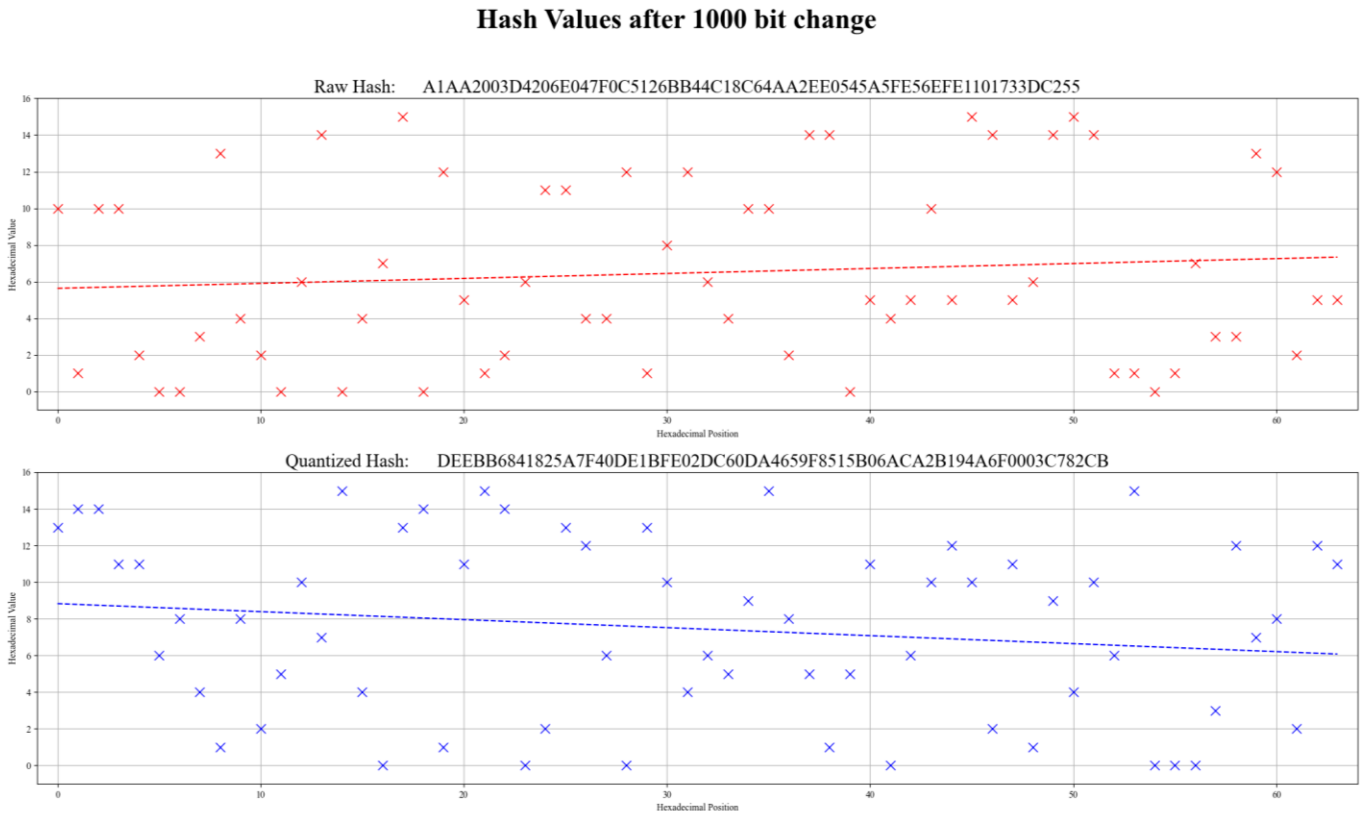


Figure 22: Hash Produced after 1,000 random bit changes

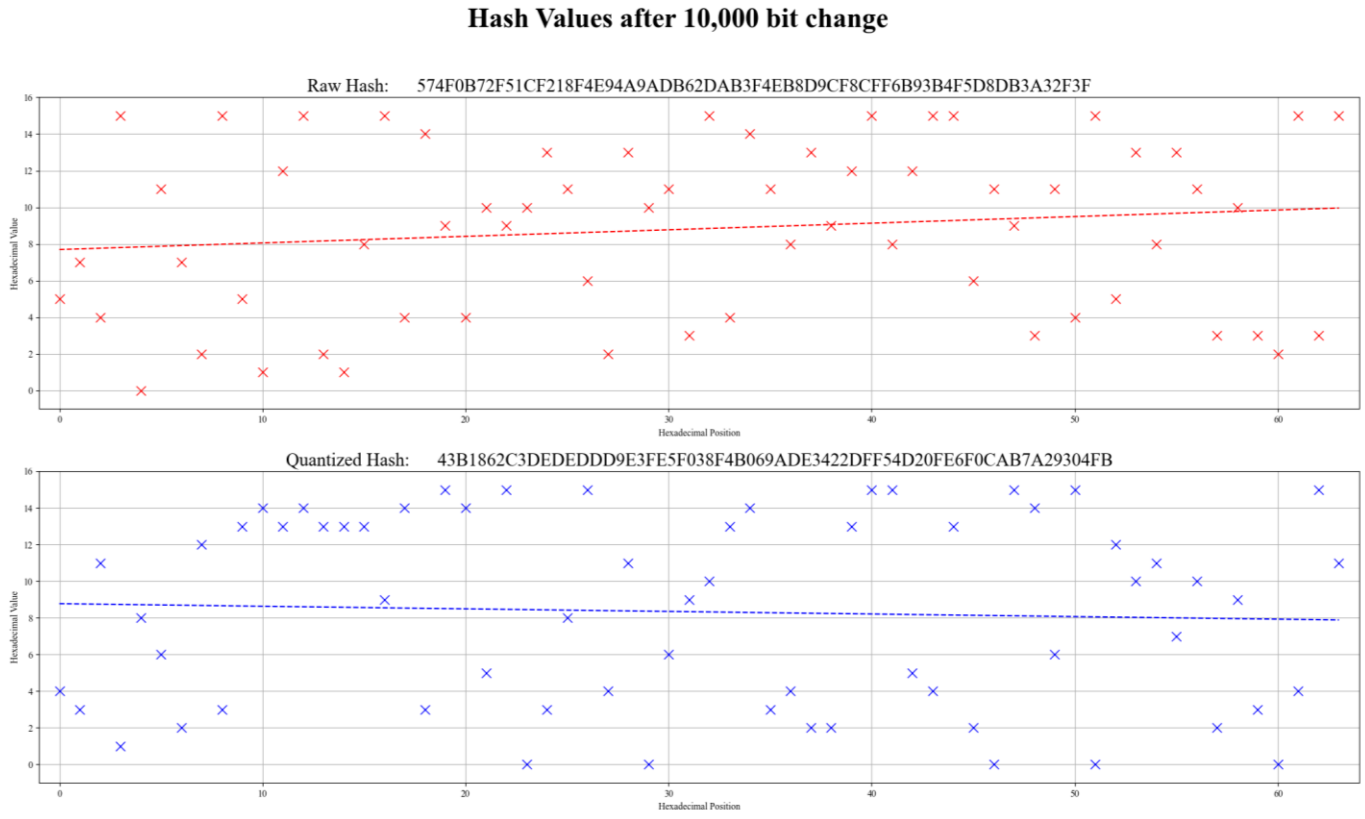


Figure 23: Hash Produced after 10,000 random bit changes

The graphs are a representation of the hexadecimal values within the hash in a base 10 format. After a single bit has been changed the hashes produced as represented in the graph are disparate, which reinforces the idea of confusion and diffusion within the avalanche effect being on display. A bit change of 10,000 bits may initially seem large, but within the context of the data being processed; 10,000 bits amounts to 7.7% of the total number of bits processed after binary conversion of the quantized EEG data, and 1.9% of the raw data. trend line was drawn on each and the gradient of the trend line was calculated. As can be seen in the graphs as well as the table shown, each graph exhibits a correlation under the threshold for either a positive or negative correlation. This confirms that the hash values are evenly distributed, among each iteration of the performed analysis experiment.

Correlation Coefficient of Each Graph

	Raw	Quantized
Original Hash	0.005357	-0.02223
1 Bit	-0.007669	-0.031685
10 Bits	0.004785	-0.019505
100 Bits	-0.004831	-0.006433
1,000 Bits	0.027038	-0.043658
10,000 Bits	0.035966	-0.014125

Figure 24: Correlation Co-efficient of Figures 18-23

❖ Collision Rate

A collision attack is a situation in which two different input messages will produce an identical hash. Collisions occur because the principle behind hashing functions is to fit a limitless combination of inputs into a fixed set of outputs, meaning at some point, two different inputs must be hashed to the same output. This is a flaw in a hash function, as collisions can be used maliciously, where an attacker could forge a fraudulent input with the same hash as that of an expected input. If this was in the case of user authentication, the attacker could gain unauthorized access to a user's account because the text input in the password field would not be that of the user's actual password, but the system would only check if the hash were the same. The data produced for the analysis of the hashing function is a by-product of the analysis of confusion and diffusion. When performing the analysis in which a random single bit is flipped a number of time to measure the Hamming distance, there were a number of instances in which a bit was flipped but there was a Hamming distance of 0, indicating that the hashes

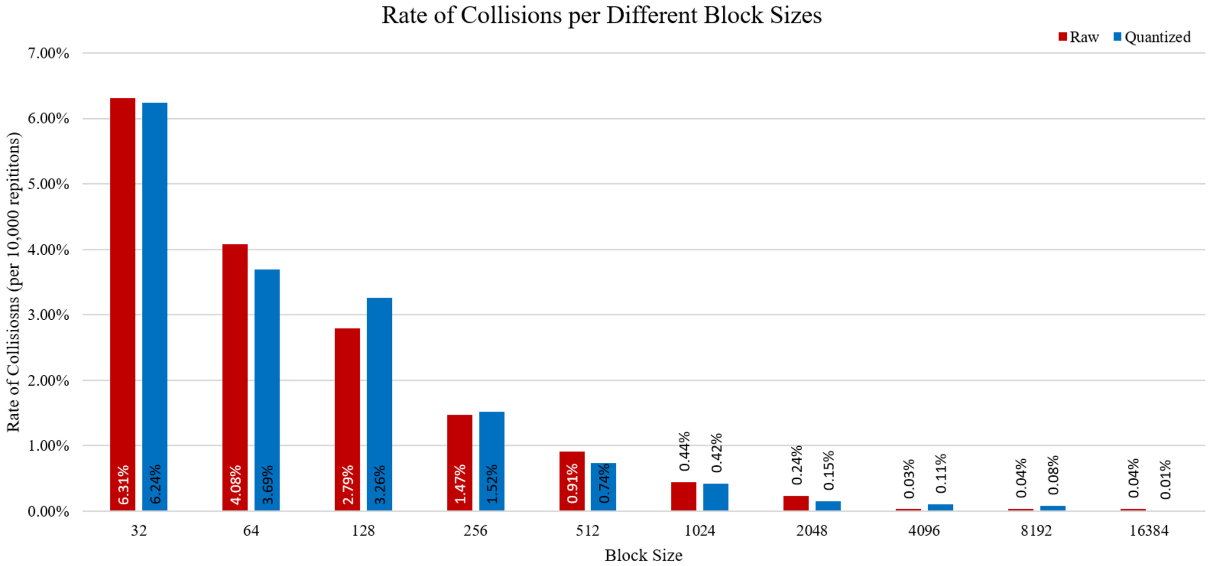


Figure 25: Rate of Hash Collisions

are identical, and a collision had occurred. The number of collisions were tallied and are presented as percentages out of the 10,000 iterations performed on each block size.

The results of the graphs show that the hash function produces an alarmingly high number of collisions. The rate of collisions begins very high with a block size of 32 and reduces with each increase to the block size, flatlining at around 4096 bits. This shows a large security flaw in the architecture of the solution as even a single collision in 10,000 iterations is a cause for concern. This runs contrary to the results presented in the previous section analysing confusion and diffusion, because that analysis focused on the trends of the results, opposed to this analysis on the outliers. In the case of security, it is crucial that the outliers are accounted for because they are the vectors in which malicious users will attack to break the system.

The hashing function MD5 has a collision probability of $1.47\text{x}10^{-29}$[69]and is seldom used today because of vulnerability to collision attacks. The more secure, and widely used, SHA-256 function has a collision probability of $4.3\text{x}10^{-60}$[69]. This means that the solution presented is much too weak to be used in any use case where security is mandatory. Considering the ideal use case would be in user authentication via EEG data, the solution fails dramatically in this regard. In order to remedy this, the program code would need to be refactored with a more sophisticated approach. A much stronger emphasis would need to be placed on the function’s level of confusion in order to make each singular change in bits even more significant in the hash produced.

❖ Distortion Cost

As the EEG data is being quantized, there will be a cost in the accuracy of the data in rounding values to the nearest quantization level. As explained, this is to the benefit of saving memory, which will increase computation times as well as account for minor differences when attempting to recreate a mental image for authentication. To calculate the exact cost of quantization, the squared-error distortion formula; $d(x, \bar{x}) = (x-\bar{x})^2$ was calculated for each of the readings within an EEG input file. The mean of which were calculated and for different levels of quantization.

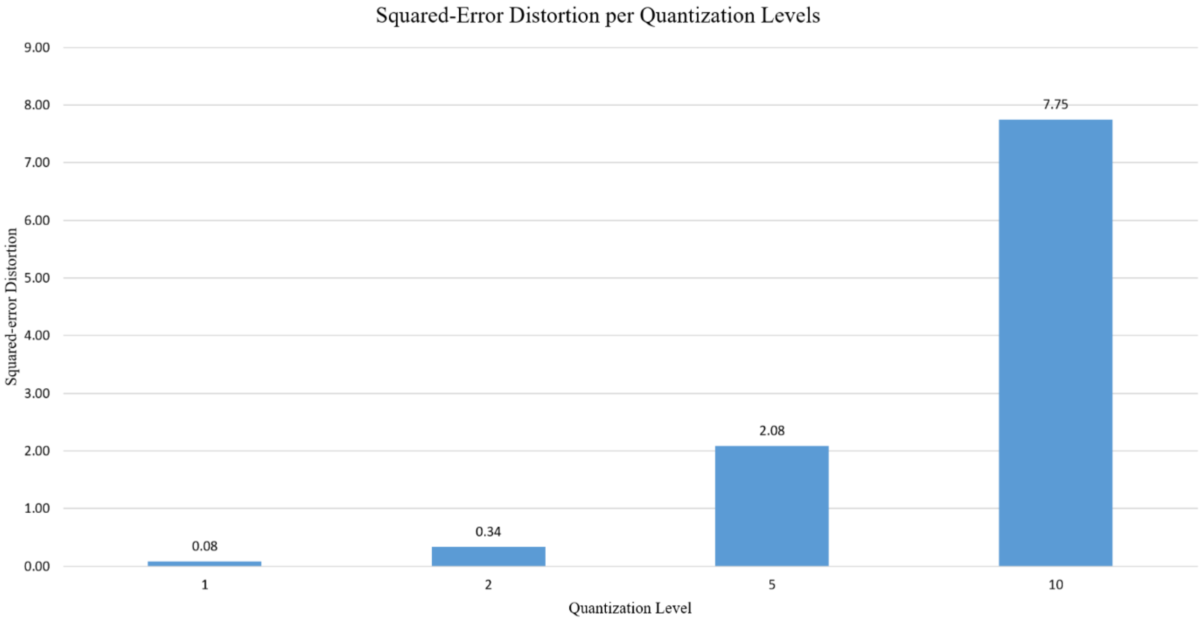


Figure 26: Squared-Error Distortion Bar Chart

The graph shows that squared-error distortion levels stayed low for quantization levels of 1 and 2, however because expensive for higher levels. This indicates that there is a limit at which too much rounding will deform the EEG values to a high degree. Ideally lower levels of quantization are preferred, because not only does the distortion of the values increase with higher quantization levels, but more aggressive rounding could also result in a higher chance of collisions because less of the detail within the data would be retained. Because it was not possible for the algorithm to be applied to an experiment on a real user or a simulation, it is difficult to determine what levels of quantization will be needed to properly account for the inconsistencies when trying to recreate a mental image of a user's "pass-thought".

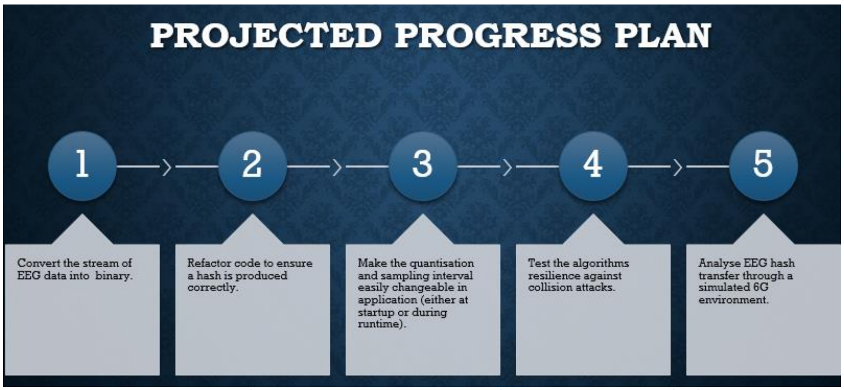
VII. Project Planning

The development cycle of this project went very smoothly. This can be attributed to the agile approach to the project development. The agile manifesto was popularized in 2001 as an alternative solution to traditional software development frameworks such as waterfall. The manifesto is represented in these four statements[70]; [1] Individuals and interactions over processes and tools, [2] Working software over comprehensive documentation, [3] Customer collaboration over contract negotiation, and [4] Responding to change over following a plan. The first and third statements were not too relevant to this development cycle as the author was really creating this project for themselves, as opposed to a product for a consumer to purchase, and since it was a solo project so there were not many opportunities to interact with other individuals. One could make the argument that the project supervisor, Dr. Barros, could be seen as a customer in a sense, as he was providing feedback on the direction of the project and which features should be more important than others. The fourth statement was definitely adhered to as there was no concrete plan to development. There were only abstract problems that needed to be solved, there was not much planning in the grand scheme of the project. As the program developed, different solutions were tried, failed, and the re-tried from a different approach as required. In this project, the second statement was the most impactful, as the author heavily emphasised always having working functionality. The author considers themselves to a novice to coding, but they have still experienced the stress of opening a project in development, only to be greeted with a list of errors. This is something that was to be avoided at all costs, as trying to debug code written on a previous day (even if it is one's own) can prove excess hassle. This is why testing was completed after every change made to the code. For risky large changes, the whole file was copied, and changes made in the copy of the file, therefore if the newfound changes brought errors, there was always an instance of the project that was working correctly. Another important aspect of agile methodology is the use of project management tools. For this project, Jira was used to record progress made. Jira proved to be invaluable, as a clear visual record of all work that has been, and is to be, undertaken was useful in characterising the progress of the project so far. Although somewhat clumsy to use for an individual project as opposed to a group project, its role in the calm development should be acknowledged.

Software development risks pose real threats to the projected productivity of a development cycle and should be identified and mitigated accordingly. To prevent any risk of data loss, all work of this project was stored in four different locations. The first being the hard drive within the authors personal computer, as most development was undertaken on it. The second was a personal USB stick of the authors, used to store any work that had been done away from the authors' home. The author would use the University of Essex's campus computers at time, and another copy of the work was stored on the University's cloud servers. The last copy was stored GitLab. Four separate locations may have been excessive, but no chances were going to be taken as lost progress is a critical setback in development. It also proves to me a big demoralising factor, possibly degrading the quality of future work output. Poor quality, buggy code is also a development risk. To combat this, code specific to a particular functionality was developed and tested in isolation. This meant that any bugs were strictly a result of implementation issues, narrowing down the number of issues and where they could be found. Scope creep is where the initial vision of a project continues to grow, including more features than was originally intended. This can become an issue because the expectations of what is to be delivered can exceed what realistically can be achieved. The inclusion of extra features can come at a cost, as the result may be a product full of half-baked ideas, instead of a solid, functioning product. Scope creep was managed by keeping expectations within the time and technical skill available to the author. This meant the removal of features that would be nice, but not integral to the core functionality of the program. One such idea was that of being able to interact with the program via a graphical user interface. This would have wasted a lot of development time without providing a lot of value. This was abandoned in place of a command line interface. The risk of failing to meet deadlines was remedied with weekly meetings with Dr. Barros. This meant that there was a continuous flow of progress, as the meetings acted as "soft deadlines", encouraging that there was a reasonable amount of advancement on the project on a weekly basis.

The momentum of this project was generally maintained very smoothly. Because agile frameworks emphasise always having a functional product, the author decided it best to divide the problem as a whole into a series of sub-problems. The main problem being the project goal of "How can one produce a hash from an instance of EEG data stored in text files?". This was divided into the following sub problems of: *"How can only relevant data from the text file be retrieved and stored for processing?"*, *"How can a series of chaotic variables be created?"*, *"How can the chaotic variables be processed with the EEG data to provide confusion and diffusion?"* and *"How can variable length data be compressed into a fixed length string?"*. This meant that core program functionalities could be developed in isolation. Then it would be tested to ensure that the solution worked as intended. Once this was confirmed, the solution is stored in its own file. This is beneficial for three reasons, the first being that doing so avoids the common issue of developing large portions of functionality in a single go, making effective debugging strenuous because a plethora of bugs are presented at once. Secondly, when making changes to functionality that was not working as intended, the author was not put into a position of being forced to fix the bugs to reclaim overall program functionality. Instead, the author could simply re-implement the standalone, functioning code, and re-attempt a correct implementation method. The third benefit is that at most points in development, there were multiple problems that need to be tackled at once. Although this may be seen as a disadvantage, this meant that if a specific problem was proving to be perplexing, there were other tasks that could be done. Therefore, there was always another manner in which to be productive, instead of wasting time mulling over the same issue. The phenomena of struggling to find answer to a problem, leaving it to do other tasks, only to return to it with more insight is called incubation[71]. The author felt it's benefits at multiple points throughout development, which they attribute to the agile development. Scheduled weekly meetings with Dr Barros, meant that there was someone to hold the author accountable for any progress (or lack thereof) during development. The author would discuss work that had been undertaken in the previous week, the current state of the project and the direction it would be best to move in. These meetings effectively acted as scrum meetings, encompassing sprint retrospectives and sprint plannings from an agile framework. The only time in which development was slowed was then the author did not progress development over the Christmas holidays. Reflecting on this presents an argument on two sides. One can argue that such a break is required to maintain enthusiasm for the project by taking time away to prevent burnout. However, the author believes that they may have fallen victim to a common development risk of overestimating the amount of time left before the deadline. Although the author is proud of the state of the project, they acknowledge that it is worthy of criticisms in some respects. This leaves the author wondering how much more improved the program could have been if the rate of development was consistent throughout its cycle.

Adaptation to change during development was not a prominent issue. This may have been due to the large degree of freedom in which the problem could have been solved. There were no third parties to demand functionality to be included or removed. Therefore, the only instances of change needing to be made was when an initial solution did not work correctly, so changes were made to the approach to succeed. This may have included altering parts that were working in isolation so that they fit into the whole program properly. For example, a planned feature of the program was to be able to read the text files and collect the number and names of the channels the readings are from. This solution would mean that the program would be more flexible, as EEG readings range from a single channel to as many as 256. Implementing this feature was more difficult than first anticipated, and considering the time constraints, the author had to adapt to these circumstances and decided to statically pre-define the channels instead. This meant that the program would only function with readings from 64 specific channels in a specific format, forgoing the flexibility of working with different collections of readings. The scope of the project had to be altered to fit the skillset of, and time available to the author.



PROJECTED PROGRESS PLAN

1. Convert the stream of EEG data into binary.
2. Refactor code to ensure a hash is produced correctly.
3. Make the quantisation and sampling interval easily changeable in application (either at startup or during runtime).
4. Test the algorithms resilience against collision attacks.
5. Analyse EEG hash transfer through a simulated 6G environment.

Figure 27: Description of a projected project plan for the second half of development.

Overall, the author is very pleased with the development of this project as it went incredibly smoothly. The image on the right is a projected progress plan, created during week 11 of term, just before Christmas holidays. At this point in time, some of the core program functionality was working, such as the tent map generation and first iteration of the compression function. The author had thought that the projection may have been ambitious, and that development would have not been so simple. In reality, development went exactly as outlined in the project plan. There were no risks or lack of technical ability that cause a major setback. This is reflective in the Cumulative flow diagram from Jira, where (especially in the early stages of development,) progress was frequently recorded and incremental. The long period of stagnation from December is that when development was paused over the Christmas and New Years holidays. Once development resumed in January, a majority of the functionality was complete. Meaning the tasks left were less frequent and much broader, typically revolving around proper implementation and project testing and analysis.

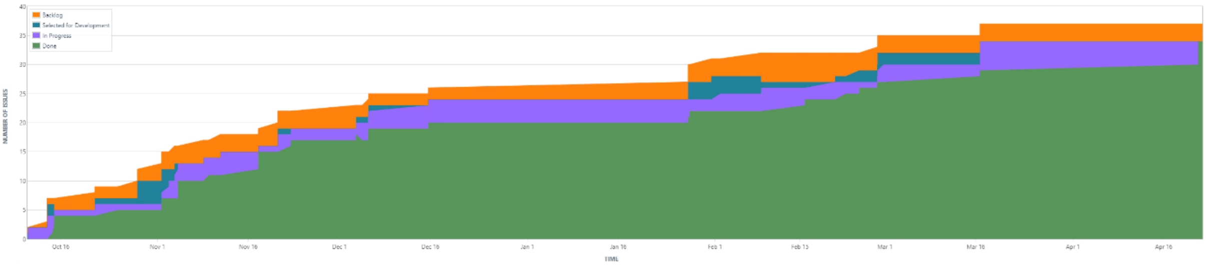


Figure 28: Cumulative Flow Diagram of project development
Source: Jira

Although a relatively small project (only hovering around 500 lines of code), it was the biggest solo project the author had created, so there were a lot of insightful takeaways from the experience. The first being that a solid approach to a project is crucial. Had the framework to development been more rigid and inflexible, this could have increased the project’s exposure to risk factors. The author thoroughly believes in the agile manifesto, as it proved its benefits in this solo endeavour, but could be even more beneficial in a team environment, where communication and organization are much more difficult. Admittedly, there were points in development where the author slowed down progression, this was due to the belief that the project was in good spot and did not require much attention at that moment. This is a fault, as when reviewing the project to write this report, there are examples of inefficient and confusion code, as well as desirable features that time did not allow for. Had the effort contributed maintained of a high velocity, there would have been more time to spare to refactor and improve the project.

VIII. Conclusion

The central aim of this study is to further the research of how Electroencephalogram data can be used as a method of biometric authentication in the future. This goal was divided into three project aims. The primary goal being to "Produce a method of using chaos theory to create an effective hash from electroencephalogram data to provide authentication required for transmission over a high throughput wireless network". This goal was partially fulfilled because a function for EEG data hashing was produced. The function was successfully able to parse a text file containing EEG data, incorporate chaos theory into its architecture via a tent map function, and produce a corresponding message digest to uniquely identify the input file. The algorithm displays desired properties of a hash function by being able to compute the hash quickly. The program is able to work with a dataset of variable length per channel, assuming that the input file contains 64 channels. As well as the program producing a hash of a fixed length of 256 bits. In this regard, the project is a success. However, the performance of the solution was not able to be simulated in a high throughput wireless network as intended because of time constraints, therefore the veracity of the solution in this regard remains unknown. The first of the secondary supporting goals was defined to "Evaluate this solution's security effectiveness". In this regard, the solution displays some success in showing some level of confusion and diffusion displayed and a reasonable level distribution of hash values. However, this is overshadowed by the solutions failure in its resistance to collision attacks. In this state, the algorithm is much too vulnerable for any security application, especially biometric authentication. Although the algorithm produced in this project is not suitable as a biometric authentication method, the research undertaken proves to show that such a solution is possible, and is a crucial pre-requisite for the progression of the possible applications that this technology can present.

The final project aim was to "Evaluate the overall effect Brain Computer Interface technology may have on society". This was covered extensively in this paper's second section of the literature review. To summarize the ideas presented; BCI's and brain wave authentication have a certain future in society. Research into the technology will continue and its possible applications further explored. Humanity can see two distinct futures ahead of itself. The first is where widespread BCI adoption will benefit those in medical need, with people being able to monitor, predict and mitigate multiple health issues from the comfort of their home. As well as providing both able and disabled users with another method in which they can interact with Internet of Things based technology. The second is one where such technology is abused by government and corporations in which people are harvested of personal data to further the agendas of thought policing and unethical neuromarketing practices. These scenarios are not mutually exclusive, but in order to prevent the latter, pre-emptive precautions must be taken for the safety of the public. This includes accurate education of the populous as well as stringent laws and regulations surrounding ethical frameworks for BCI design.

With regards to the weaknesses of this research project, one could criticize that the implementation of chaos theory was not required, and its inclusion could have hampered the potential of this project. Given that there are plenty of reliable hash functions that do not implement any chaotic features, it is a reasonable conclusion to say that its implantation in this project may have done more harm than good. Especially considering the solutions lacklustre security features, had the author focused on designing a more logically reliable and sound algorithm, then perhaps the results presented would show a better performance. However, because there are multiple examples of secure hash functions being developed that do include chaotic features, the author concludes that the implantation of chaos in-and-of-itself was not an inherently flawed idea. Instead, the mistake came from the author heavily prioritising a based chaos first, as opposed to a cryptographically secure function. The choice to include chaos should have been made after a better evaluation of the underlying logic of the algorithm, the authors skillset, and the time available for the project

Time was a large limiting factor of the depths this endeavour could be explored. This is due to the fact that there was an inflexible, hard deadline of the CE301 module. Although it should be acknowledged that the author is also at fault for not planning and managing time as could have best been done. Therefore, there are a list of improvements that could be made to ameliorate the work achieved in this project. The biggest being to refactor the algorithm to rectify the security flaw of such a high collision probability. The author believes that this flaw is in part due to a contrived method of permuting the binary values of the EEG data. Had a permutation box been implanted, this could have solved for such as issue. The second largest issue is that of the format requirement of input files being restrictive. The program is designed around a specific format of representing EEG data. In this, it means that the program is not flexible, as trying to use a file with any number of channels that is not 64 will not work as intended. Considering that EEG readings range from 1 to 256 channels, a method in which the program can adapt to how many channels are presented would be more a better design. Reviewing the program code shows areas of inefficiency, there are sections of the code which may be unclear for someone to understand the flow of the program. Therefore, the code could be refactored to remove a lot of redundant commands and functions. The program will prompt the user for which version of the SHA algorithm to use in creating the hash, however SHA-3 seems to be an improvement upon SHA-2, meaning there is not much reason to select the latter. This makes this feature redundant and can be removed. Other improvements are more minor features, such as creating a graphical user interface with a file explorer feature to make interacting with the program simpler. Being able to select multiple files to hash and compare simultaneously would be a welcome addition as well.

IX. References

[1] A. Rowe, “Study Reveals Average Person Has 100 Passwords,” *tech.co*, Nov. 09, 2021. https://tech.co/password-managers/how-many-passwords-average-person

[2] A. Shukla, “Why Did Humans Evolve Pattern Recognition Abilities?,” *Cognition Today*, Oct. 06, 2019. https://cognitiontoday.com/why-did-humans-evolve-pattern-recognition-abilities/

[3] NordPass, “Most common passwords of 2020,” *nordpass.com*, 2021. https://nordpass.com/most-common-passwords-list/

[4] B. Meyer, “Most common passwords: latest 2022 statistics,” *CyberNews*, Apr. 04, 2021. https://cybernews.com/best-password-managers/most-common-passwords/

[5] A. K. Jain, A. Ross, and S. Prabhakar, “An Introduction to Biometric Recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, Jan. 2004, doi: 10.1109/tcsvt.2003.818349.

[6] Office For National Statistics, “Internet users: 2015,” *Office for National Statistics*, May 22, 2015. https://www.ons.gov.uk/businessindustryandtrade/itandinternetindustry/bulletins/internetusers/2015

[7] Office for National Statistics, “Internet users, UK: 2020,” *Office for National Statistics*, Apr. 06, 2021. https://www.ons.gov.uk/businessindustryandtrade/itandinternetindustry/bulletins/internetusers/2020

[8] D. Buil-Gil, F. Miró-Llinares, A. Moneva, S. Kemp, and N. Díaz-Castaño, “Cybercrime and shifts in opportunities during COVID-19: a preliminary analysis in the UK,” *European Societies*, vol. 23, no. 1, pp. 1–13, Aug. 2020, doi: 10.1080/14616696.2020.1804973.

[9] T. S. Rappaport *et al.*, “Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond,” *IEEE Access*, vol. 7, pp. 78729–78757, Jun. 2019, doi: 10.1109/access.2019.2921522.

[10] E. Klein, “Informed Consent in Implantable BCI Research: Identifying Risks and Exploring Meaning,” *Science and Engineering Ethics*, vol. 22, no. 5, pp. 1299–1317, Oct. 2015, doi: 10.1007/s11948-015-9712-7.

[11] J. J. Shih, D. J. Krusienski, and J. R. Wolpaw, “Brain-Computer Interfaces in Medicine,” *Mayo Clinic Proceedings*, vol. 87, no. 3, pp. 268–279, Mar. 2012, doi: 10.1016/j.mayocp.2011.12.008.

[12] C. Klaes, “Chapter 28 - Invasive Brain-Computer Interfaces and Neural Recordings From Humans,” *ScienceDirect*, vol. 28, pp. 527–539, Aug. 2018, doi: B9780128120286000288.

[13] E. Klein and J. Ojemann, “Informed Consent in Implantable BCI research: Identification of Research Risks and Recommendations for Development of Best Practices,” *Journal of Neural Engineering*, Jun. 2016, doi: 10.1088/1741-2560/13/4/043001.

[14] S. Waldert, “Invasive vs. Non-Invasive Neuronal Signals for Brain-Machine Interfaces: Will One Prevail?,” *Frontiers in Neuroscience*, vol. 10, no. 295, Jun. 2016, doi: 10.3389/fnins.2016.00295.

[15] D. A. Craig and H. T. Nguyen, “Adaptive EEG Thought Pattern Classifier for Advanced Wheelchair Control,” *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2007, doi: 10.1109/iembs.2007.4352847.

[16] V. P. Buch *et al.*, “Network Brain-Computer Interface (nBCI): An Alternative Approach for Cognitive Prosthetics,” *Frontiers in Neuroscience*, vol. 12, no. 790, Nov. 2018, doi: 10.3389/fnins.2018.00790.

[17] R. Matthews, “On the Derivation of a ‘Chaotic’ Encryption Algorithm,” *Cryptologia*, vol. 13, no. 1, pp. 29–42, Jan. 1989, doi: 10.1080/0161-118991863745.

[18] INMOS Bristol, “IMS T800 Architecture,” *www.transputer.net*, Jan. 1988. https://www.transputer.net/tn/06/tn06.html#:~:text=The%20Inmos%20transputer%20family%20is

[19] Tech PowerUp, “AMD Ryzen Threadripper 3990X Specs,” *TechPowerUp*. https://www.techpowerup.com/cpu-specs/ryzen-threadripper-3990x.c2271

[20] M. Amin, O. S. Faragallah, and A. A. Abd El-Latif, “Chaos-based Hash Function (CBHF) for Cryptographic Applications,” *Chaos, Solitons & Fractals*, vol. 42, no. 2, pp. 767–772, Oct. 2009, doi: 10.1016/j.chaos.2009.02.001.

[21] ScienceDirect, “Message-Digest Algorithm 5 - an Overview | ScienceDirect Topics,” *www.sciencedirect.com*. https://www.sciencedirect.com/topics/computer-science/message-digest-algorithm-5#:~:text=MD5%20is%20the%20Message%20Digest

[22] J. Vidal, “Toward Direct Brain-Computer Communication,” *Annual Review of Biophysics and Bioengineering*, vol. 2, no. 1, pp. 157–180, Jun. 1973, doi: 10.1146/annurev.bb.02.060173.001105.

[23] T. Elbert, B. Rockstroh, W. Lutzenberger, and N. Birbaumer, “Biofeedback of Slow Cortical Potentials,” *Electroencephalography and Clinical Neurophysiology*, vol. 48, no. 3, pp. 293–301, Mar. 1980, doi: 10.1016/0013-4694(80)90265-5.

[24] A. R. Murguialday *et al.*, “Brain-Computer Interface for a Prosthetic Hand Using Local Machine Control and Haptic Feedback,” *IEEE Xplore*, pp. 609–613, Jun. 2007, doi: 10.1109/ICORR.2007.4428487.

[25] C. Chen, C. Lin, and S. Ming, “Hand Orthosis Controlled Using Brain-computer Interface,” *Journal of Medical and Biological Engineering*, vol. 29, pp. 234–241, Nov. 2008, [Online]. Available: https://www.researchgate.net/publication/225304561_Hand_Orthosis_Controlled_Using_Brain-computer_Interface

[26] G. Pfurtscheller, C. Guger, G. Müller, G. Krausz, and C. Neuper, “Brain Oscillations Control Hand Orthosis in a Tetraplegic,” *Neuroscience Letters*, vol. 292, no. 3, pp. 211–214, Oct. 2000, doi: 10.1016/s0304-3940(00)01471-3.

[27] B. Rebsamen *et al.*, “Controlling a Wheelchair Using a BCI with Low Information Transfer Rate,” *IEEE Xplore*, pp. 1003–1008, Jun. 2007, doi: 10.1109/ICORR.2007.4428546.

[28] K. Tanaka, K. Matsunaga, and H. O. Wang, “Electroencephalogram-based Control of an Electric Wheelchair,” *IEEE Transactions on Robotics*, vol. 21, no. 4, pp. 762–766, Aug. 2005, doi: 10.1109/TRO.2004.842350.

[29] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, and K. Diserens, “An Efficient P300-based Brain–computer Interface for Disabled Subjects,” *Journal of Neuroscience Methods*, vol. 167, no. 1, pp. 115–125, Jan. 2008, doi: 10.1016/j.jneumeth.2007.03.005.

[30] J. R. Wolpaw and D. J. McFarland, “Control of a two-dimensional Movement Signal by a Noninvasive brain-computer Interface in Humans,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 51, pp. 17849–17854, Dec. 2004, doi: 10.1073/pnas.0403504101.

[31] D. J. McFarland, D. J. Krusienski, W. A. Sarnacki, and J. R. Wolpaw, “Emulation of Computer Mouse Control with a Noninvasive Brain–computer Interface,” *Journal of Neural Engineering*, vol. 5, no. 2, pp. 101–110, Jun. 2008, doi: 10.1088/1741-2560/5/2/001.

[32] T. A. Kayagil *et al.*, “A Binary Method for Simple and Accurate two-dimensional Cursor Control from EEG with Minimal Subject Training,” *Journal of NeuroEngineering and Rehabilitation*, vol. 6, no. 14, May 2009, doi: 10.1186/1743-0003-6-14.

[33] A. J. Doud, J. P. Lucas, M. T. Pisansky, and B. He, “Continuous Three-Dimensional Control of a Virtual Helicopter Using a Motor Imagery Based Brain-Computer Interface,” *PLOS One*, vol. 6, no. 10, p. e26322, Oct. 2011, doi: 10.1371/journal.pone.0026322.

[34] D. J. McFarland, W. A. Sarnacki, and J. R. Wolpaw, “Electroencephalographic (EEG) Control of three-dimensional Movement,” *Journal of Neural Engineering*, vol. 7, no. 3, p. 036007, Jun. 2010, doi: 10.1088/1741-2560/7/3/036007.

[35] C. Guger, H. Ramoser, and G. Pfurtscheller, “Real-time EEG Analysis with subject-specific Spatial Patterns for a brain-computer Interface (BCI),” *IEEE Transactions on Rehabilitation Engineering*, vol. 8, no. 4, pp. 447–456, Dec. 2000, doi: 10.1109/86.895947.

[36] D. C. Irimia, R. Ortner, M. S. Poboroniuc, B. E. Ignat, and C. Guger, “High Classification Accuracy of a Motor Imagery Based Brain-Computer Interface for Stroke Rehabilitation Training,” *Frontiers in Robotics and AI*, vol. 5, no. 130, Nov. 2018, doi: 10.3389/frobt.2018.00130.

[37] Z. Gao, T. Yuan, X. Zhou, C. Ma, K. Ma, and P. Hui, “A Deep Learning Method for Improving the Classification Accuracy of SSMVEP-Based BCI,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3447–3451, Mar. 2020, doi: 10.1109/TCSII.2020.2983389.

[38] F. Gilbert, C. Pham, J. Viaña, and W. Gillam, “Increasing brain-computer Interface Media depictions: Pressing Ethical Concerns,” *Brain-Computer Interfaces*, vol. 6, no. 3, pp. 49–70, Jul. 2019, doi: 10.1080/2326263x.2019.1655837.

[39] K. Chan-soo, “A high-tech, brain-shrinking Future,” *KoreaJoongAngDaily*, Jun. 12, 2009. https://koreajoongangdaily.joins.com/news/article/Article.aspx?aid=2906013

[40] N. Perkins, “5G and Brain Tumours,” *Brain Tumour Research*, Mar. 13, 2020. https://www.braintumourresearch.org/media/our-blog/blog-item/our-blog/2020/03/13/5g-and-brain-tumours

[41] E. Klein, “Informed Consent in Implantable BCI Research: Identifying Risks and Exploring Meaning,” *Science and Engineering Ethics*, vol. 22, no. 5, pp. 1299–1317, Oct. 2015, doi: 10.1007/s11948-015-9712-7.

[42] J. del R. Millán and J. M. Carmena, “Invasive or noninvasive: Understanding brain-machine Interface Technology,” *IEEE Engineering in Medicine and Biology magazine: the Quarterly Magazine of the Engineering in Medicine & Biology Society*, vol. 29, no. 1, pp. 16–22, Jan. 2010, doi: 10.1109/memb.2009.935475.

[43] DataReportal, “Global Social Media Stats,” *DataReportal*, Apr. 2022. https://datareportal.com/social-media-users

[44] B. Dean, “Social Network Usage & Growth Statistics: How Many People Use Social Media in 2022?,” *Backlinko*, Oct. 10, 2021. https://backlinko.com/social-media-users

[45] S. Lesaja and X.-L. Palmer, “Brain-Computer Interfaces and the Dangers of Neurocapitalism,” *arXiv:2009.07951 [cs]*, Sep. 2020, doi: 10.48550/arXiv.2009.07951.

[46] A. Kahraman and AytekinP., “A New Research Approach in marketing: Neuromarketing,” *Journal of Management Marketing and Logistics*, vol. 1, no. 1, pp. 48–62, Mar. 2014, [Online]. Available: https://dergipark.org.tr/en/pub/jmml/issue/32448/360837

[47] D. J. Kuss, M. D. Griffiths, L. Karila, and J. Billieux, “Internet Addiction: a Systematic Review of Epidemiological Research for the Last Decade,” *Current Pharmaceutical Design*, vol. 20, no. 25, pp. 4026–4052, 2014, doi: 10.2174/13816128113199990617.

[48] N. Xanidis and C. M. Brignell, “The Association between the Use of Social Network sites, Sleep Quality and Cognitive Function during the Day,” *Computers in Human Behavior*, vol. 55, no. A, pp. 121–126, Feb. 2016, doi: 10.1016/j.chb.2015.09.004.

[49] J. Fox and J. J. Moreland, “The Dark Side of Social Networking sites: an Exploration of the Relational and Psychological Stressors Associated with Facebook Use and Affordances,” *Computers in Human Behavior*, vol. 45, pp. 168–176, Apr. 2015, doi: 10.1016/j.chb.2014.11.083.

[50] K. W. Müller, M. Dreier, M. E. Beutel, E. Duven, S. Giralt, and K. Wölfling, “A Hidden Type of Internet addiction? Intense and Addictive Use of Social Networking Sites in Adolescents,” *Computers in Human Behavior*, vol. 55, no. A, pp. 172–177, Feb. 2016, doi: 10.1016/j.chb.2015.09.007.

[51] M. Koc and S. Gulyagci, “Facebook Addiction among Turkish College Students: the Role of Psychological Health, Demographic, and Usage Characteristics,” *Cyberpsychology, Behavior, and Social Networking*, vol. 16, no. 4, pp. 279–284, Apr. 2013, doi: 10.1089/cyber.2012.0249.

[52] I. Wolniczak *et al.*, “Association between Facebook Dependence and Poor Sleep Quality: a Study in a Sample of Undergraduate Students in Peru,” *PLOS One*, vol. 8, no. 3, p. e59087, Mar. 2013, doi: 10.1371/journal.pone.0059087.

[53] A. Przepiorka and A. Blachnio, “Time perspective in Internet and Facebook addiction,” *Computers in Human Behavior*, vol. 60, pp. 13–18, Jul. 2016, doi: 10.1016/j.chb.2016.02.045.

[54] N. S. Hawi and M. Samaha, “The Relations among Social Media Addiction, Self-Esteem, and Life Satisfaction in University Students,” *Social Science Computer Review*, vol. 35, no. 5, pp. 576–586, Aug. 2016, doi: 10.1177/0894439316660340.

[55] R. A. Elphinston and P. Noller, “Time to Face It! Facebook Intrusion and the Implications for Romantic Jealousy and Relationship Satisfaction,” *Cyberpsychology, Behavior, and Social Networking*, vol. 14, no. 11, pp. 631–635, Nov. 2011, doi: 10.1089/cyber.2010.0318.

[56] I. Pantic, “Online Social Networking and Mental Health,” *Cyberpsychology, Behavior, and Social Networking*, vol. 17, no. 10, pp. 652–657, Oct. 2014, doi: 10.1089/cyber.2014.0070.

[57] B. J. Maiseli *et al.*, “Brain Computer Interface: Future, Challenges, and Potential Threats,” *Biomedical Signal Processing and Control*, Apr. 2022, doi: 10.2139/ssrn.4073630.

[58] Meta (Facebook), “BCI milestone: New Research from UCSF with Support from Facebook Shows the Potential of brain-computer Interfaces for Restoring Speech Communication,” *Tech at Meta*, Jul. 14, 2021. https://tech.fb.com/ar-vr/2021/07/bci-milestone-new-research-from-ucsf-with-support-from-facebook-shows-the-potential-of-brain-computer-interfaces-for-restoring-speech-communication/#:~:text=Established%20in%202017%2C%20Facebook%20Reality

[59] Google, “Brain Team – Google Research,” *Google Research*. https://research.google/teams/brain/

[60] J. Tan and A. E. Tan, “Business under Threat, Technology under Attack, Ethics under Fire: the Experience of Google in China,” *Journal of Business Ethics*, vol. 110, no. 4, pp. 469–479, Oct. 2012, doi: 10.1007/s10551-012-1494-0.

[61] B. Light and K. McGrath, “Ethics and Social Networking sites: a Disclosive Analysis of Facebook,” *Information Technology & People*, vol. 23, no. 4, pp. 290–311, Nov. 2010, doi: 10.1108/09593841011087770.

[62] F. Yousefi, H. Kolivand, and T. Baker, “SaS-BCI: a New Strategy to Predict Image Memorability and Use Mental Imagery as a brain-based Biometric Authentication,” *Neural Computing and Applications*, vol. 33, pp. 4283–8297, Aug. 2020, doi: 10.1007/s00521-020-05247-1.

[63] F. Yousefi and H. Kolivand, “Brain Signals as a New Biometric Authentication Method Using Brain-Computer Interface,” *Encyclopedia of Computer Graphics and Games*, pp. 1–14, Aug. 2019, doi: 10.1007/978-3-319-08234-9_370-1.

[64] Science Daily, “Youth Adapt Faster than Seniors to Unexpected events, Study Finds,” *ScienceDaily*, Jan. 18, 2011. https://www.sciencedaily.com/releases/2011/01/110118113453.htm#:~:text=Reaction%20in%20older%20adults

[65] A. Allen, “Young Adult Perception and Acceptance of Biometric Technology,” *Youngstown State University*, Nov. 2012, doi: 849518589.

[66] H. Begleiter, “EEG Database,” *kdd.ics.uci.edu*, Oct. 13, 1999. https://kdd.ics.uci.edu/databases/eeg/eeg.data.html

[67] K. Boyini, “How to do twos complement on a 16-bit signal using Python?,” *www.tutorialspoint.com*, Apr. 19, 2018. https://www.tutorialspoint.com/How-to-do-twos-complement-on-a-16-bit-signal-using-Python

[68] B. Preneel, “Davies–Meyer Hash Function,” *Encyclopedia of Cryptography and Security*, pp. 136–136, 2005, doi: 10.1007/0-387-23483-7_96.

[69] G. Ramirez, “MD5: the Broken Algorithm,” *Avira Blog*, Jul. 28, 2015. https://www.avira.com/en/blog/md5-the-broken-algorithm

[70] Agile Manifesto, “Manifesto for Agile Software Development,” *Agilemanifesto.org*, 2001. https://agilemanifesto.org/

[71] S. M. Smith and S. E. Blankenship, “Incubation effects,” *Bulletin of the Psychonomic Society*, vol. 27, no. 4, pp. 311–314, Apr. 1989, doi: 10.3758/bf03334612.