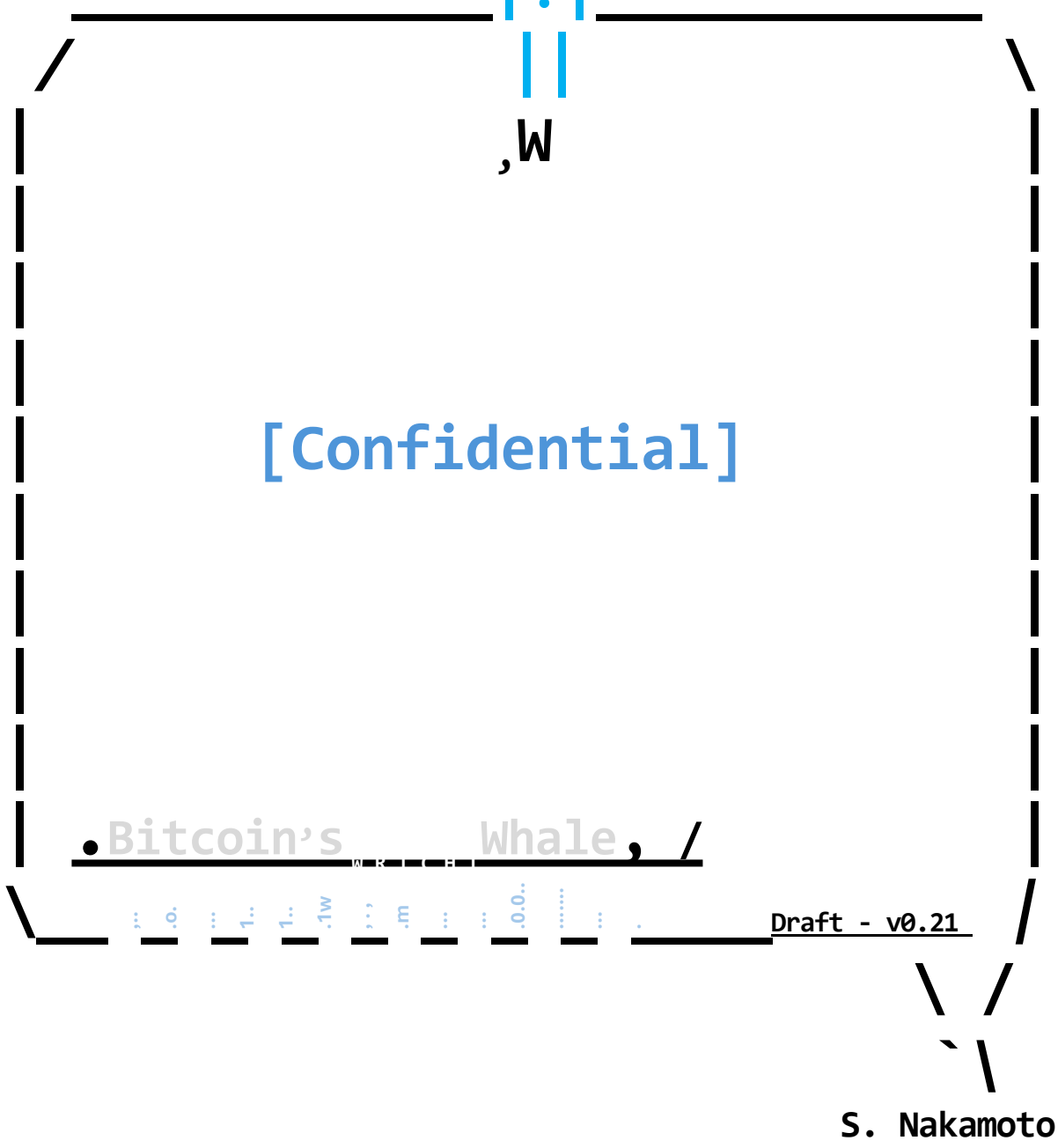
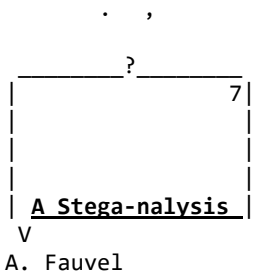
10/03/2025





A. Fauvel

Contents

Motivation

There have been comments in court that certain individuals have proof they are the author of the Bitcoin Whitepaper that are not cryptographic in nature. This information came to light during the questioning of such an individual in an administrative high court in the United Kingdom about the identity of Satoshi Nakamoto. At the time of this comment Steganographic techniques were being discussed. While details were not shared this requires further investigation. If steganographic methods have been used on the PDF via software or otherwise, it must leave traces that we could possibly follow and hope to unmask its author once and for all.

Apologies for the lengthy proof, however, this being such a large world "mystery" I thought it best to document the method as I performed it, I then after satisfied I had the correct method went over the document adding context and reordering some of my findings to round of any rough edges and bring a cohesive story through that the reader can understand. Although we make a lot of assumptions throughout, our result must answer at least some of our questions about the authorship to be convincing and utilise known features that can reasonably be interpreted by someone skilled or practiced in decoding steganography. Steganography's main security feature is that the naïve readers are unaware that it has been used. If one were to simply go looking for it appropriately, they would likely find it were it would be logical to use it. Most people don't scan every document looking for signs of manipulation. As such can be transmitted publicly without detection by the public. This does however mean that people maybe able to decode it and unwilling or unable to discuss it openly. This report is aimed at breaking down this language barrier allowing the reader to understand how we got to our result and why.

While I give my own conclusions at the end, they are merely my opinions which should not be looked at as bias but merely the right motivation to go looking and then publicise my findings.

Steg-analysis

It shouldn't be too hard to believe that people wish to communicate secretly within or without set groups. Traditionally people assume complex cryptographic systems are the best form of secret communication. However, many of these complex mathematical systems fail in a few areas in which people may wish to use it. The main failure from a secret message senders' perspective is plausible deniability or surveillance by unknown eavesdroppers today or in the future.

In a cryptographic system there is a set protocol that is widely known. When this protocol is applied to a message that we wish to investigate it typically requires a private key to be used at the same time. This key can either be retrieved or generated by parties depending on their relationship, for good cryptographic systems an enemy shouldn't be able to independently generate either part of the key within a reasonable time frame. Ideally until the end of the universe without being too cumbersome to use day to day.

The art of steganography though, while it achieves a similar goal to cryptography is an entirely different process. In steganography, while a basic framework is utilised that is widely known, the exact methodology is kept as a protected secret. The concealment method must use a host message. This host message will dictate the medium the message can take and thus, the type of rules we might be able to create to conceal it.

You should now understand why Steganography is fundamentally different to cryptography. It is a messy art but creates order out of something that looks chaotic. You must realise though, that you are only reading these words in this order because that is what you have been told to do. It's what the author expects you to do as they wrote it for the purpose of being read in that way. This mutual protocol in language is the only way to transmit a message common understanding.

A truly secret message has not context to which it can be related, however, nothing exists in isolation, thus a selectively secret message with meaning can not be prevented from being exposed to the public. The only mitigating action once exposed is to deny, which if done correctly is plausibly believed.

In the 16th-century a German author, Johannes Tritheism adviser to a number-of emperors and a "humanist" published a 3-part book titled "Steganographia". The first 2 books were relatively unremarkable although highly coveted cryptography books, however, book 3 was quite the deviation. This was what appeared to be a spell book of occult magic. Written in Latin the Roman Catholic Church quickly band it. Its title is Greek, 'Stega' meaning covered 'o-graphia' meaning writing. Fitting that the author would title a book on hidden writings in a language that it is not written in. Also, fitting that the third book in the trilogy was completely misunderstood as we can assume was the author's intention since it was used as a cover. Steganography is a language hidden within a language. This has been proven to be the case by two independent European researchers. If you want to know what the message is, you will have to look elsewhere as I have not read it myself.

From a fundamental perspective, in cryptography you would be able to write cypher text on a blank sheet of paper. For steganography you need a host message or medium so that the

message is concealed from view and thus classified as hidden. If you were to apply steganography to a blank sheet of paper there are two main methods, either you mark the paper in other ways and then add your message to it by manipulating the marks. This is best done throughout their creation to mean something other than what they may first appear, or you make them such that the marks can't be seen on the paper at all. This is a subtle but fundamental difference between crypto and stega-nography. This fundamental difference in mechanisms and set up requirements changes the way they are used. Cryptography is typically reserved for personal privacy or communications, while Steganography is used for rights management and secret communications with a cover. Most critically cryptography uses lots of mathematical armour, steganography uses "smoke and mirrors" (misdirection) as a core architectural feature.

The application of rights management makes logical sense if you understand that steganography requires two messages occupy the same space. One that is seen, and one that is selectively revealed. The reveal is referred to as a 'decode' in modern day stega' parlance. This dual delivery mechanism allows those that wield the skill to deliver content throughout society creating an information asymmetry. The commercial application of this technique today is digital rights management (DRM) software. Its model's security relies on the honesty of participants. The content is embedded with uniquely identifying information that is intentionally difficult to detect and/or remove effectively without seriously degrading the information.

A digital watermark might look like a company logo in the corner of the screen or transparently over the centre of media content, this is a non-secret example of how it is used to deter people from acting dishonestly. A more advanced system creates "noise" within the image which is imperceivable to the naked eye but can be decoded even if the data is damaged or compressed.

This works by creating something like a QR code and making the difference between white and black a very small difference in colour grade. In other words, reduced contrast until, to you and me it, just looks Gray, only a computer can tell the different. This unique information is then transparently overlayed as tiles on top of the host content in various sizes and over lapping such that even if cropped or distorted one of the "QR-codes" will likely be detected and decoded. These elements could be the size of a single pixel or a significant percentage of the screen.

Today, we haven't reached the level of traceability, where everything delivered online uses such a digital rights management technique. With the speed at which the computation industry is developing, it might not be too long. However, we must always consider the application of technology when deploying it, sometimes it is mission critical, other times it is a nice to have, other times it is a nuisance that is unnecessary. Whatever the case, the uses of technology can always be positive and negative depending on how they are deployed. For normal people what we are about to attempt is like magic or brute forcing the solution we want. Yet, taking something as ordinary as a spam email or academic paper, and pulling something we recognise from it in a way that is hard to believe, just has a method that is not understood. It's not madness just a miscommunication. To many this is a dark art. So logically speaking, a dark and mysterious character like Satoshi maybe well-versed in. They seem to be a world class expert in every other we have seen them utilise. This is well documented as no matter how hard people may try, all attempts to identify him or them remain inconclusive.

More clandestine uses are to deliver content to the public and secret information to a small group of insiders. This could either be to communicate stock trades, confidential information by informants or plans for world domination. In summary this darker side of steganography could probably appropriately be called a technology of spies and criminals.

A common way people use steganography without realising it is to make intentional spelling errors to convey meaning other than what the words themselves read. Most commonly “U” in place or “you” or the less common IYKYK [If you know, you know]. A combination of emojis complimenting text can also be considered a form of steganography, directing readers to wherever the author most desires. If someone can read a word in context with the spelling error a reader may be able to infer what is being communicated. Corn or something like |*orn communicates the meaning without necessarily using all the completely correct characters. People very commonly use this kind of intentional manipulation to reduce the size of a message or, when hiding, to avoid surveillance, censors and filters.

The author having just decoded an intricate message embedded in an email from 1997 on the CypherPunk Mailing List (CP-ML) using steganography feels confident that they can spot intricate intentional errors that indicate and infer instructions. After all the CypherPunks (CP) claim that Satoshi is one of them and that Bitcoin is their own invention, so maybe it will also provide some insight on who authored it, if only we look at it in a new way. The puzzle on the mailing list is quite simple in comparison to what we should expect from an academic like paper, errors in a paper are a lot less acceptable in academia and would arouse suspicion.

Although this decode is very different from the ones found on the CypherPunk Mailing List it is believed that the principles used will likely be the same. Steganography while unique every time it is applied follows some core principles which allow a method of communication which takes extraordinary effort to interpret and refine into a message that the either party is confident in. In effect steganography is a technique that can take longer considerably longer to read that it did to write, cover not included. Making a puzzle out of a picture simply requires cutting it with a die, putting the puzzle together again is something of a slow, interpretation consideration time, trial and error, refinement process. With each piece fitting together one by one.

One of the most ancient subsets of steganography, “Text-Form-matting, Erorr”, or TFE is used extremely frequently on the CP mailing list and are intended to look like spam adverts. However, they are not your run of the mill spam mails the public is used to receiving. Through the lens of steganalysis they are highly targeted adverts sent to a unique audience who have a high chance of converting even with the additional work it takes to decode them. If you want to compare steganography with today’s technology each message is like a little computer program written in its own language where an out of place comma is the difference between a cover and the message the messenger genuinely wished to send. This is partly why the people that most use it has an in depth understanding of code. Of course, not all coders have been sending secret messages to one another, there is just a significant amount of overlap in skill sets.

As you should know by now, you need a cover to start creating a message using steganography, as such steganographic methods are typically employed when finalizing a document. The author choosing which errors to keep along the way. Looking for any additional messages or hints that they may be able to sneak past regular reader but might be noticed by a keen

investigator. As such it doesn't take that long to create a puzzle such as this one, nowhere near as long as it takes to solve them. A puzzle this complex while it took a lot of time to consider, probably took a few hours or days to implement depending on the exact methods used and how complete our final solution ends up being, if we can tell.

While it does happen frequently in practice, it is not advised to create a cover that is not genuine to send a message, this would cause too much suspicion for investigators that you don't want to suspect anything. An unnatural cover that needs to be generated doesn't have enough links to the real world to be believable. A covert spam email advertising a criminal racket for example, would need a real business or person for the organisation to point to if the investigators are going to be assured nothing else is really happening. To be truly covert, and from a design perspective, sometimes less is more pleasing and ideal for the application. Ambiguity leading to many possible paths. Yet, this ambiguity must not be so unmanageable and overwhelming that it no longer has meaning to the reader no matter which side of the informed vs uninformed aisle they may fall. Both messages need to be understood by their respective parties for the method to be effective. Ideally with little to no cross contamination between the groups.

Being a steganography puzzle, what we are looking for must relate to the cover used.

Written under a pseudonym, leaving the world so many questions, ideally are looking for something related to an identity such as a name, date, place or organisation. To get this message out of the text, we will be searching for a method of extraction which relates to the information we are reading about, not necessarily the information we are looking for. The first signs we will be looking for are plausibly deniable errors that are on the edge of acceptable. These artifacts will also not be obvious but must be present.

Finding a method that might yield a meaningful message seems like a very daunting task. Especially if the method is complex, the instructions are probably also complex. The author would have had to covertly enter them into the paper as entire sentences or paragraphs. The number of logical leaps we will have to take while might be numerous should, if in theory and practice used consistently throughout be limited by the medium and the message. As a famous youtuber once said, for any problem there exists a solution with a finite number of problems to fix. Once we are finished there should be significantly fewer mistakes in the paper, if any at all. This depends on the skill of this secretive author and messenger.

If this message is to be confirmed in any meaningful way it must be related to the subject matter and events surrounding the puzzle. When we achieve our result, (of course we came to one else you wouldn't be reading this) given the size of the Bitcoin Problem the only way to judge if our answer is correct is to compare it with real-world events. The paper being like a carefully delivered time capsule, cannot be changed in any way since it is widely distributed by technology enthusiasts. The technology it describes relying heavily on the inability of third parties to easily alter information once it has been widely distributed in public which because of this those enthusiasts expect any analysis to be done on files that are (hash) identical to the ones Satoshi published. For our purposes, the file doesn't have to be hash identical. A simple PNG or JPEG or printout of each page will do for the most part. Virtually all our evidence will be visible to the naked eye allowing anyone with a copy in its original form to verify the method we are utilizing. We do have some rudimentary copy and pasting to do to be effective in

documenting what we are doing but that's about it. To put any concerns to rest, all artifacts we have analysed are on the gold standard of papers. Bitcoin.org/bitcoin.pdf

If you want to follow along with a paper, you source yourself then the hash you need to use to check you are using the most UpToDate original and genuine copy released by Satoshi Nakamoto himself is:

```
b1674191a88ec5cdd733e4240a81803105dc412d6c6708d53ab94fc248f4f553
```

Over the course of writing this report it has been noted that several publications and individuals refer to something known as "Entropy similarity" or similar. This in principle is a steganographic method, however, unless one can pinpoint the relationship that steganographic analysis unveils this is a "handwave" method of distracting the audience from pinpointing why a relationship might exist.

The output of this report will be divisive no matter the output that we have come to. To address this concern, it is the authors intent and documentation process has been to look upon each feature and character and carefully consider what it might represent with hard-hearted objectivity. This requires taking the most likely route for someone sending secret messages at each junction assuming they are. We should get a pretty good idea of weather or not the things we are seeing are random with some basic statistical and local analysis quite early in the analysis, it is from this point that those that disagree with the result will find themselves looking for wholes to poke. I invite these people to dispute everything they find with as much evidence from within the paper itself. They then must link that to the events of the world since its publication. The goal we have here is to detail each anomaly or interesting feature seen then follow the most interesting ones which have the highest likely hood of being intentionally manipulated by an author looking at unspoiled document. Once these areas have been exhausted move on to the next most interesting ones or wherever the information, we may find is leading us. Like a discovery game we're playing with the original author and/or publisher or an exploratory miner looking for fractures, in the earth, big and small which might contain some mineral or resource.

Fresh eyes

The Bitcoin Whitepaper is an incredibly clean document. Academic in nature, concise and precise. The mark of a real expert in their field who may have had some aggressive editing assistance. The language is a mix of extremely simple words and industry specific jargon but communicating complex ideas with a brevity that sticks to facts, and ONLY facts.

Formatted meticulously in all aspects it has been read by millions of technical people for supposedly endless hours each. Investigators going so far as to travel the world in search for its mysterious author. The author clearly suspecting their invention had struck a new kind of digital mine and that this document would be read by millions of people the world over quickly vanished. So, it might be likely that in crafting this paper they took extreme care, more care than anyone might suspect at the time.

In terms of formatting consistency is an issue that permeates the document, the font size of the abstract is different to that of the body of the document and that creates the possibility that each section of the paper has its own unique font size. Immediately this could be evidence of a highly sophisticated computer-generated steganography algorithm, but this is beyond the scope of this paper. While noted, as we proceed, we won't let it influence us at this early stage in our investigation.

There are no spelling errors and arguably no grammatical ones either at first glance. Spellcheck does point to some unideal wordings, missing or inconsistent hyphens and commas in non-ideal places, but nothing that seems out of place. At least none that would suggest to anyone not looking that there might be steganography hidden within it at first glance. Suspicion is often the first step in looking at something with fresh eyes, so following our gut we can suspect that someone's ego wouldn't let them change the world in such a manner without being able to prove they are the ones behind it. At least in a way that is plausibly deniable unless you already know the people involved. Maybe they wanted a fail-safe that they could show to allies they meet along the way or maybe an enemy pinpointing one of the players and causing them to live in the chaos. This is a battle for the worlds communication and accounting system after all. Not a small task and if you were to take on such a monumental task, the people around you might call you just a little crazy, which might in fact cause you to go a little crazy.

This meticulous effort extends to the diagrams having embedded text and vector-based graphics. Not just pixel-based images so often found in regular commercial word processing software. This maybe because our author wanted to restrict a researchers focus to the more rudimentary text and visual based method of TFEs.

The metadata lists OpenOffice 2.4 which was released in March 2008 as the producer of the PDF; however, we can't know if a document was produced in another program first and then loaded into Open Office for some additional functionality and then exported to PDF or some other combination thereof. An auditable trail of this editing process hasn't been securely recorded anywhere that we know of it may not even exist. Trusting this record is also a problem that only the invention the document describes could hope to fix, leaving us with a chicken and egg paradox. The inventor could even be demonstrating the capabilities of his solution with an

example of a problem society faces which his invention could fix. If only it was used to its full capabilities.

We must also note that this may not be the last piece of software to modify the content of the document. Software like SNOW can be used on a finished document to add additional information while retaining all other properties such as metadata and human readable content such that the information can only be extracted with a key. It does this by using the negative dead space at the ends of the lines to add whitespace characters such as spaces and line breaks. The steganographed document can then be loaded into the same software and a password entered to extract a human readable message. While it is worth noting if someone wants someone else without any contact or knowledge to eventually find their name hidden directly within the document, we may have to look for errors we can see visually to understand a direction to follow and then pick up clues as to what to do next.

Re-view Document

The document is 9 pages and 12 numbered sections long.

Each sections title (page number) and points of interest:

0. Abstract (1)
    - Smaller font
1. Introduction (1)
    - Nothing of interest
2. Transactions (2)
    - Diagram
    - Reference [1]
3. Timestamp Server (2)
    - Diagram
    - Usenet {Capitalisation choice}
    - Reference [2-5]
4. Proof-of-Work (3)
    - Diagram
    - Adam Back named
    - Reference [6]
    - Usenet {Consistent capitalisation}
    - SHA-256 and zero bits {Missing hyphen?}
    - Hyphenation usage in voting mechanics {Consistent}
5. Network (3)
    - 6 item list with 7 lines and 2 paragraphs below
    - Duplications across lines with possible structure
    - Possible hints in paragraph below
    - Use of only semi colon in the document outside of calculations
6. Incentive (4)
    - No visual anomalies
7. Reclaiming Disk Space (4)

- References [7],[2],[5]
- Diagram
- Possible words and letters aligning to indicate something
- Calculations, numbers, date, RAM

8. Simplified Payment Verification. (5)
    - Diagram
9. Combining and Splitting Value. (5)
    - Diagram
10. Privacy. (6)
    - Diagram
    - Use of “tape” in quotation marks
        i. Only use of quotation marks outside of the references
11. Calculations. (6)
    - Use of numbers and equations
    - Reference [8]
12. Conclusion. (8)
    - No visible artifacts
13. References (9)
    - Names and dates in a list 1-8

Additional notes:

1. Double spaces used after period
    1.1. Signs of an older typist.
2. Formatted as justified throughout
    2.1. Possibly to hide inconsistent white spaces for steg-software usage.
3. Usage of hyphens seems a little excessive but generally consistent.
4. Section 7 has a 1-6 list
5. There are 7 diagrams
    5.1. None have boarders or captions
    5.2. 2 in the same diagram have a border and caption, segmenting them from each other.
6. There is code and mathematical equations on pages 6-7.
7. Few references throughout, yet when used, many are used at once.

Error in-depth

There is one clear mistake in this document from a formatting perspective that I didn't point out in the list but to me in a document such as this points to intentional manipulation. It is a mistake that is arguably not a mistake without knowing the specifics of the references and conventions. The section this error is contained in is written out below. On page 4 under the 7th section Reclaiming Disk Space is just 2 short paragraphs long:

7. Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.

[Diagram of pruned and unpruned Merkle tree]

A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

Before discussing the main error lets address the numbers within the calculations in the second paragraph. Given that Bitcoin uses a 10-minute block time and an 80-byte header it is unlikely these numbers are used in a steganographic manner. These are the numbers we would expect given the subject medium is Bitcoins defining document. Given that these are the real numbers it is unlikely that an inventor would choose these variables to communicate in a little side quest after the invention. Similarly, the numbers indicating the amount of RAM a computer has, or the year of authorship is not in fact controlled by the author they are simply world facts and thus is unlikely to be used or manipulated to convey a secret message.

The error we will investigate from here are the references listed in the order [7],[2],[5]. This is unusual as the convention would be to put them in numerical order regardless of relevance to the point being referenced or chronology of them in the references. Typically, if one needs multiple references for a single point they would be in numerical order, the relevance to the point itself is irrelevant and subjective.

References should by convention be used when directly related to the point being made. This point of the transactions being hashed into a Merkle tree is not unique to this paragraph. The question we should ask ourselves is why the author felt compelled to bundle all three together in this point. It is the first time a tree structure is mentioned but that doesn't mean every mention of tree structures in the references should be referenced here. Some of these references have also been used at other times inferring the author knows the reference material well but mentioning them here a second time is again unnecessary.

References [2-5] were used during the statement of public broadcasting like "a newspaper or Usenet post". The other time the Usenet (Network/Database) is mentioned is in section 4 near Adam Backs name and reference. S. Haber and W.S. Stornetta are the authors of 3 out of 4,

which is interesting to note. If we are looking for an author's name or initials the references are an obvious place to put them.

The references for the occasion in which we **do not** suspect manipulation are below:

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

The referencing system used by the paper is in line with the numeric references of the Vancouver formatting style following the format:

Last Name Initial. Title article: Subtitle. Abbreviated journal title. Year Month Day; volume (issue); page.

There are a few slight differences between our author and this convention, our author does the following:

Initial. Last Name, "Title article: Subtitle," Abbreviated journal title, Month Year.

Our author has elected to swap "Last name **Initial**." to "**Initial** Last Name" and the "Year **Month**" to "**Month** Year" but also to put the titles in quotation marks. All but one of these changes is common. Dates are often swapped in error as locating the publication month of articles in annual publications sometimes leads to conflicting dates. Swapping the last name and initial however is not normal after someone has gone to such lengths to conform to other standards. The order of an author's name is an international standard across all types of academic fields papers and journals. These conventions state that references are to be [Last Name] followed by [Initial.], so this is a peculiar alteration in an otherwise well close to flawless document is notable from a steganography perspective.

As anyone in academia would tell you, getting your references right and correctly formatted is of utmost importance.

The sentence with the out of order references speaks of a Merkel tree. As we would expect the R. C. Merkle's paper which describes a Merkle tree is in reference 7. Se7en is the most relevant so should come first, some people would say. Traditionally convention dictates that references should be used only when directly relevant or necessary, when multiple, they should be in numerical or alphabetical or chronological order depending on formatting choice. They key is to be consistent and conform to standard. This list of references is none of these things and not within the convention. The reference list is in order of references used which does conform to traditional conventions. So, why on this occasion do we have 3 in-line references out of order?

Darren Kellenschwiler or Deggen in the blockchain industry pointed out that these citation formats are in fact IEEE standards, which follow the format:

1. J. Ive, A. Max, and F. Yvon, “Reassessing the proper place of man and machine in translation: A pre-translation scenario,” Mach. Transl., vol. 32, no. 4, pp. 279–308, Dec. 2018, doi: 10.1007/s10590-018-9223-9.

This explains the quotation marks, order of the names and dates, but not the fact that the references are out of numerical order. These inconsistencies still go against this citation standard and as far as I have seen all citation standards because it is such a unnatural to do and see. While citation standards are informative to infer signals when errors or inconsistencies are found. Due to how similar a lot of them are, many people including me, can mistake one for another while thinking the author just made a mistake due to our understanding of what we know. Apologies if you think this most of this section was a waste of time, but I decided to keep it in the report in the interest of transparency.

Another contribution by a valued contributor Freddie Honohan, has pointed out that if this document was complied in LateX since the software handles references in a separate file called a bibtex syntax document with each entry in the format:

```
@INPROCEEDINGS{9505235,
  author={Bartoletti, Massimo and Lande, Stefano and Zunino, Roberto},
  booktitle={2021 IEEE 34th Computer Security Foundations Symposium (CSF)},
  title={Computationally sound Bitcoin tokens},
  year={2021},
  volume={},
  number={},
  pages={1-15},
  keywords={Bitcoin;Computer security;Bitcoin;tokens;neighbourhood covenants},
  doi={10.1109/CSF51468.2021.00022}}
```

These entries are then given an index number denoted by their usage in the paper, since when references are used, they are always in numerical order, thus this had to be manually corrected to get the result we are seeing of [7],[2],[5]. If Satoshi wanted this order in LaTeX they would have to set the label of the citations to:

```
\cite{ them7 } \cite{ in2 } \cite{ order5 }
```

The default would have been and seems to have been used elsewhere [2-5]:

```
\cite{cited2} \cite{cited5} \cite{cited7}
```

And the third unused option is:

```
\cite{ them2 } \cite{ in5 } \cite{ order7 }
```

The order of the references may also be an error when compiling the document through writing and editing, this is something that would likely be spotted and corrected by either correcting the order or removing the additional unnecessary references. If only for the visual and psychological satisfaction. This is the type of error that feels unnatural in a document such as this. Quickly scanning the paragraph an author looking for things to fix, would likely catch this and want to fix it. Much like a tiny dent in an otherwise perfectly flat mirror (in the eyes of its author) causing distortions that are outsized for the size of the imperfection delivers. The more information an imperfection carries the larger its distortions in the rest of the text but the bigger the text the more area for them to hide. You can only move these distortions around and make them smaller. There are limits to this without using machines, but in this analysis, we are limiting ourselves due to assumptions of what the author is trying to communicate and with whom. There is a

strong chance that on this journey of invention they thought “no-one will ever figure this out”, or “when someone figures this out, I’m either screwed or saved.” So, at the time may have seemed like a message in a bottle riding the waves of an ocean, possibly only ever between him and God.

We won’t dwell on which standard is used since it is not the information that will clearly define or tell us anything. There are many more things to cover that are less ambiguous and much more informative. Instead, we will look to apply the key to something else in the document in some way. This may or may not be the references themselves.

A Key Position

Before we consider the references themselves, let us consider the potential uses of an artifact such as the references in a specific numeric order.

It could be used as a key in several ways, not only by the numbers, but also the position of where to find elements. In this scenario if we are looking for the first element, we might find it in the second position, if we are looking for the second element it would be in the third position and third element we could find in the first position. The elements could indicate a final order or be a simple mask with a gap at the minimum, we can only speculate until we look further.

```
---
       [7][2][5]
          V          <- Starting position / Select character (Most simple/common idea)
       [3][1][2]

---
       [7][-][5]
             V       <- 2nd Position cycling (?)
       [3][1][2]

---
       [1][2][3]     <- Reorder final string (?)
---
```

It is unlikely that these three numbers are meant to be manipulated mathematically given how they were found and where they point us.

The papers themselves, while they all mention a binary or Merkle tree structure and are highly related to the paper in general they do not seem directly related to what is being referenced. Reading them it seems most of the core design for bitcoin was already established in theory but putting it into practice with proof-of-work at the top of the tree and a scripting language to control the digital signatures as leaves was a novel idea. Especially the revelation of making it a public protocol with a clear, although mysterious, starting point.

The question as to why these references are here together maybe found by considering the order of and the numbers themselves, it could be a in reference to some initials or something else. The first letter being 7 the next one being 2 and the third letter being 5. Three numbers aren't a lot of space to convey information so we may also need to consider that we haven't got all our parts for this yet.

The if these numbers are doing something rudimentary like producing the initials of the author, we simply need to find the area to apply the key to.

The first letters of the 7th, 2nd and 5th chapters are:

```
7.  Reclaiming Disk Space
2. Transactions                          [R/D/S].[T].[N]
5. Network
```

Using our theory that the first element would be found in the second position yields D, however, the other titles don't have a position to choose from. This inconsistency in application suggests we are not looking in the right place, so this is unlikely to be what we are looking for.

Most often with steganography there is little ambiguity in what actions should be taken as the manipulator makes sort of transmission errors to guide a would-be decoder. What is the use of embedding a secret message in such a way if you don't ever want it to be found. It's not a message if it doesn't make sense. There are no items that indicate that there is more manipulation to occur here.

Steganography requires a cover text that can be manipulated into a form without being so malformed that it no longer serves its purpose. A heading with no punctuation, spelling errors, formatting inconsistencies or other interesting visual features is a place that is barren to a genuine secret message. You can't hide an elephant in a desert without the dessert being relatively massive in comparison, or a whale in the ocean for that matter. What is needed is a forest with lots of little "nooks and crannies" to infer and highlight things with a series of "mistakes" that an untrained eye would overlook.

Section 7 is where this artifact was found in the first place and re-examining the same paragraph for something new to confirm which letter to use in the string produced from the section titles doesn't seem to yield anything meaningful other than a potential comma and apostrophe usage but here, they seem appropriate and without any other context cannot understand them.

Moving to the references themselves, we can see that there is a lot of punctuation white space, numbers and formatting choices within. All the relevant references in the order they are provided to us by the key are:

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

Reference 7 only has one author being R.C. Merkle, following the form of the key to get our first letter we should find it in the second position, normally this would have been the initial after a last name so the second element would have been R if we were following conventions rigorously. However, we find that the last name Merkle is in the third position instead of the first. The form of this name has three elements.

```
First initial. Second initial. Last Name,
```

If the author was following standard conventions for the format, they have chosen we would have had:

```
Last Name. First initial. Second initial,
```

Thus, swapping the names change our answer creating a point where two "mistakes" intertwine. Considering our key is also three elements in size the author would be inferring that we should take the second initial, C.

```
[C].[-].[-]
```

Following this “initial” idea, the second letter that could be provided by reference [2] could be any of the following:

```
   1   2   3                        |
1 [1],[2],[3]        H. M,[ ]       |      1  2  3  4  5  6  7  8
2 [4],[5],[6]        X. S. A,       |      H. M, X. S. A, J.-J. Q,
3 [7],[8],[9]        J.-J. Q,       |
                                (1) | (2)
```

There doesn’t seem to be anyway to choose between these numbers or methods them unless we reconsider our application of the key. Which row do we take and which character, or which number in the array do we take. There doesn’t seem to be any indications on the page as that might help solve our problem.

If we consider scenario 1, we have a matrix which we could apply our key two, equally we have an array that we could chose from. There is no, way for us to know in which configuration we should apply our suspected key to or how many times so we must not have all the information needed.

Looking at the (Design of a secure timestamping service with minimal trust requirements) paper itself , it appears that the S stands for Serret as part of a last name not an initial as the reference has lead us to believe. Other papers that use this reference used the full name instead of just an initial here are a few examples all written before the Bitcoin Whitepaper. Since we got here following errors and formatting, we should likely skip trying to figure out the key and just take this immediate error as our choice. Thus S, for the last name Serret, in the string X. S. Avila is likely the letter we should chose, but some other confirmation would be nice.

```
[C].[S].[-]
```

This error in name is most likely an error implanted by the author to informs us that we should only be taking the initials we are given, and not the initial from last names since they have been removed from the set intended by our author. This might indicate that we are not looking for a name in this section, but a set of initials.

This time if we assume that only first name initials that we are given be included as we have been “instructed” by the previous two examples. Recalling that this rule was indicated to us because the usage of the name was not established convention for this reference at the time since. S is not an initial as we were led to believe when we first looked.

It also likely also tells us that we should apply our key consistently when we have three elements in any puzzle pieces that we might find.

The third letter could be:

```
S. Haber,
W. S. Stornetta.
```

We have a lot of candidates for S, most likely happenstance or a stenographer confirming our previous choice. It is also in the second position of the second name as the previous one was. However, this time because we only have two names to choose from this time, we might infer that we only take the first name initials we are given. These two names together have three first name initials in order: [S], [W], [S].

Often, we must immediately apply what we learn from the previous step to the next one, following a path little followed in a landscaper's garden created with malicious rigour but without being done so noticeably that those that do not take such care over their process would ever even think it was done intentionally.

This may to an individual not versed in the art of steganography, this may seem like an indication that we should include last names instead of what I have interpreted. As the key brought us here, we must still consider the validity of our [7],[2],[5] key and the process it is with in. The person manipulating the information for us to interpret doesn't know the order in which we are decoding things. Although they may have something in mind due to the nature of steganography, they can't know what we try to analysis first. So, they don't know which direction we will be attacking the puzzle from unless they have given us instructions, which we haven't yet seen evidence of. If we had obeyed the rule that the key cycles when used the result may be ambiguous. All of this suggests that someone skilled in the art of steganography would not make such a mechanic without directing the user to it. With these assumptions we can feel confident that when we apply this key, we should not cycle through the numerical numbers but only consider the middle initial as part of our string.

Assuming that the steganographer is guiding us to a result with the consistency in this mechanic from the string S.W.S. yields a final string three characters:

```
[C].[S].[W]
```

--- (sidebar) since you probably definitely now know where

Found by one of our researchers and first supporters known around the blockchain space as **mohrt**, by simply counting the number of letters using the following digit of each reference, is using:

Reference line [7] take the letter in position 2
Reference line [2] take the letter in position 5
Reference line [5] take the letter in position 7

```
      2 V
[7]   R.C. Merkle
          5 V
[2]   H. Massias, X.S. Avila, and J.-J. Quisquater
                7 v
[5]   S. Haber, W.S. Stornetta
```

It is widely known that most experts overlook simple methods, while I was looking for two reasons to choose a letter, morht was doing something different but what we speculated might be the case. Once he had applied the key once to collect the references, he then went through the list and cycled through the key s number until he had to use the first number again.

The problem I have with this method is that is generates three potential results an no way to really chose between them. Those choices are:

```
             7 V
[7]    R.C. MerkLe
        2 V
[2]    H. Massias, X.S. Avila, and J.-J. Quisquater
           5 v
[5]    S. HabEr, W.S. Stornetta
```

And

```
           5 V
[7]    R.C. MeRkLe
            7 V
[2]    H. MassiAs, X.S. Avila, and J.-J. Quisquater
        5 V
[5]    S. HabEr, W.S. Stornetta
```

Resulting in either:

```
[C],[S],[W] or [L],[M],[E], or [R][A][H]
[W],[C],[S] or [E],[L],[M], or [H][R][A]
[S],[W],[C] or [M],[E],[L], or [A][H][R]
```

This is nine different possibilities, only one of which could be correct. We need a reason to chose one or the other. That reason is embedded somewhere in the mistakes or formatting choices, it maybe that if we apply the key in several logical ways, we will find one string of characters is reproduced more often but we still can't rely on that as proof. We need to look in other areas that will confirm or conflict with this.

This contribution helped me explain the way I did things a lot more clearly and while this all these things may be by chance, much of this so far, looks to be designed.

Upon learning this I then checked a few other things. The S in W.S. Stornetta, stands for Scott, and is also part of the last name, not the middle name. This has not affected our result, and it is also the way that he references himself so maybe Satoshi just trying to be consistent with how he treats last names with two parts. Nevertheless, these are formatting choices indicating we have the string we are looking for, even if we aren't certain which it is.

So far all of this may look like coincidence or an investigator looking for an answer he wants. I'll remind the reader that I didn't chose the 3 numbers. They are an artifact that cannot be changed or influenced by me, only what I chose to do with it. There are logical things to do with it was we have explored, and illogical things to do with it, which we have not. This has a high likely hood of an extremely skilled steganography, trying to tell his readers that no matter how you would logically apply the information in front of you, all paths lead to him. Creating as many paths as possible to lead people of many different skill sets to a single common result. Wanting the message to be found. This is common in a lot of steganography applications, since you want the person trying to find a message to find it. When steganography is done correctly many things come together to create what would seem to be just a number-of unimaginable coincidences when in fact it is a message that has been designed, by an individual or a group.

Craig Steven Wright is a claimant to Satoshi. This very strongly suggests that we don't have to rearrange the letters. While this is quite the remarkable coincidence 3 letters are roughly a 1 in 17,000 chance of occurring which while quite unlikely to be by chance isn't beyond the realm of possibility that this is just random chance.

R.A.H are the same initials used the email address of Robert Hettinga, one of the most prolific posters of the CypherPunks mailing list. Although, I would like to get into the details of certain members of this group that collectively claim the identity of Satoshi, the particulars of this on this group that proudly lives in the shadows is largely outside the scope of this report. However, to give you an idea of who RAH is, after news of Tim Mays death in 2018 it Robert Hettinga seems to have lost interest in keeping up his postings and began signing his last emails with an equals sign possibly indicating that he was ill and also coming to the end of his life as the equals points to nothing. He could also simply be planning on retiring his activities. Another possibility is that he didn't exist and is someone else, or it is a piece of software responding to certain people's commands. That last point might be going too far for a lot of you, but if you want to find out more about the people involved in the Cypherpunks then visit investigation7 on patreon.com and support the investigation for more insights and get your name as a supporter in the book that is being written to shine a light on them.

Given the importance and sensitivity of the Bitcoin White Paper, I feel it is unlikely that we will find the authors full name steganographic embedded within it. Although may have not known just how important it would be to some people, before launch everything must have only been flashing thoughts of potential outcomes. Writing out their full name would have likely been seen as a risk too great to take, especially given that the paper was first published to a mailing list of cryptographers.

The problem with using this as proof of anything is that it is plausibly falsely affirmable. In other words, a leading question or directing the witness. If we give someone the opportunity to affirm knowledge of something without them having that knowledge before being asked, then we give away the game. What could be done is ask them for their proof and cross check their claimed method with our own, however this would be premature. It is possible this is simply the checksum of the message we need to decode. In steganography decoding you typically find the checksum first, learn a few rules, then find the main puzzle that results in a message. You then know if you decoded the full message correctly because it matches what you already know.

When breaking cryptography while you are guessing random numbers until you find one that matches, like what we are going to be doing next. You can't break a cyphertext without knowing something else about it such as what encryption algorithm what used, to make the breaking easier it is useful to know what kind of message you are expecting. During the second world war it is now quite widely known that one of the weaknesses in the enigma algorithm is how it treated duplicate letters. The Nazi party starting messages with a weather report and ending messages with "HH" allowed code breakers to guess the days encryption key much more rapidly than they otherwise would have. It is this weakness in cryptography that Steganography also treats with care as it is used to infer instructions as part of the algorithm, typically to remove elements as we would do in a cryptographic system.

The Playfair Cypher specifically doesn't allow duplicate letters as part key since it create conflicts in within the key table that is created. Further, if your message has duplicate letters and they are within the same digrams [blocks of 2 letters] then we must substitute the second instance with the padding character chosen, typically X or Q. When we notice duplication of words, letters or other elements this may indicate that there is something to remove or

something to manipulate. It is difficult to tell what it is exactly you need to do without knowing a lot of other things about the puzzle you are facing. So, it may require looking in a similar area or back to where you came from to have an indication of what you should do next. This is going to be important to know moving forward as our puzzle becomes more complex.

After quite a bit of debate with a few supporters and contributors mainly motrh and Fredderic Honohan about why we would use ROT1 (rotate one) and be expected to get the correct characters without a signal as to why to do this. While speaking with Fredderic it occurred to us that the references are the error signal itself is ROT1 [7][2][5] is ROT1 of [2][5][7] thus fixing the error. Using the key in that form on the refferences is the most correct steganographic process to follow. Incredibly simple if you know.

I'm not satisfied that these initials are expected to be the proof, and I don't expect the read is either, I also don't expect our mysterious author does either. So far there is no way of knowing if the author intended for this to be used this way, without asking them directly and if they answer would potentially bring an end the puzzle before we have exhausted all possible avenues. There are several other elements in the paper that we should investigate, given that we have just decoded an individual's initials we may find that the full decode includes the same initials again, an initial and a name or a full first middle and last name.

Checksum of dis'

With all this considered it might be quite likely that this is a check sum of some sort, meaning that we are not done with our decode. It is common in steganography decodes to start with the checksum and then move to the main puzzle, this is because in steganography it is easier to move forward if you know roughly what you are looking for. The document is quite large to be used as a cover, giving ample opportunity to hide messages throughout. A skilled steganographer would likely want to incorporate many pages and chapters in a steganography proof. Ideally all of them in at least some small way, if possible, but remaining modular such that if the paper was split up each section together may still be used as proof, although in this scenario is unlikely as each section is quite small.

While we could investigate our summary more let us first contemplate some of our learnings so far, the key [7],[2],[5] found in section 7, the initials and dates swapped around and the addition of quotation marks around the title.

Quickly scanning our summary, we can see that we noted that chapter 10. Privacy has the word "tape" within quotation marks. The titles of the references are also in quotation marks, this could indicate that we perform some kind of manipulation to the title based on the word "tape".

The references quotation marks are:

```
"b-money,"
"Design of a secure timestamping service with minimal trust requirements,"
"How to time-stamp a digital document,"
"Improving the efficiency and reliability of digital time-stamping,"
"Secure names for bit-strings,"
"Hashcash - a denial of service counter-measure,"
"Protocols for public key cryptosystems,"
"An introduction to probability theory and its applications,"
```

There are no significant formatting choices or errors within these titles, the choice of comma before the end of quotation marks is standard for in Vancouver reference list formatting. The choice of Capitalisation for the first letter of the title except for b-money where it is a choice of the b-money author doesn't seem to be of significance; however, we do know from the history of the document that the b-money reference was altered after the document was finished to correct a date and list its webpage. Early versions of the document while could be located isn't necessary for our purposes since the author made the revisions and republished the document themselves. Unlike most other documents though, millions of hash identical copies have been distributed all over the world and even on the blockchain itself. It is one of these copies that we are using as the recognised original file.

The information surrounding our data point of "tape" is all about privacy pointing out that the times and sizes of trades is information that is made public but who the parties are isn't revealed. Topical of course. If we remember the dates of the references are also in the incorrect order as we mentioned in our comparison of the Vancouver reference list format convention. We may be able to use the word tape as a key to extract some more information the date area as we did with the initials and potentially as indicated by any formatting error. The dates of each reference are:

```
[1] 1998.
[2] May 1999.         [May is 5th month]
[3] 1991.
[4] 1993.
[5] April 1997.       [April is 4th month]
[6] 2002.
[7] April 1980.       [April is 4th month]
[8] 1957
```

Remarkably the references we found out of order, are also the ones with months included in the dates. This would suggest much more strongly that the key we extracted is correct and applied to the correct area. However, this could still be by chance. Other papers that cite the material in this list of references have months in their entries, so why chose or accidently use them in only these three papers. If not for an expert steganographer using elements sparingly, yet appropriately to elegantly point any would-be investigator in the right direction in a few different ways. The choice of putting the quotation marks around such a word would suggest that we are to read the dates in some order as a key of some kind. The author safe in the knowledge that if we can't figure this puzzle out, we still can't get it from scanning the document for something related to this which would bring us to the reference error we caught.

If we place the months in alphabetical order, we have a tie between 5 and 7 as they were both published in April. Ordering these two by date we start with 7.

```
[7, A, 80] = > R.C. Merkle,                              April  1980
[5, A, 97] = > S. Haber, W.S. Stornetta,                 April  1997
[2, M, 99] = > H. Massias, X.S. Avila, and J.-J. Quisquater,  May    1999
```

Doing this ends up giving us a chronological order. However, this seems to give us double confirmation of what we have assumed is the wrong order. This again doesn't match the order of the key we found.

If we didn't find the key the steganographer was likely betting that we might be able to figure out that M might stand for middle. Giving the same order they are used on page 5. If the order had been [7],[5],[2], being chronological ordering it would have also been unusual and worth investigating. That is because using references should be in chronological order when using the Harvard standard, but this document as discussed is using the Vancouver Standard. In other places such as in section 3 where 4 references are used [2-5]. These references are also not in chronological order but in the order, but the order they appear in the reference list which is also not in chronological order but as per convention. Taking a quick look at the dates of these [2-5] refferences:

```
[2] May 1999.
[3] 1991.
[4] 1993.
[5] April 1997.
```

This inconsistency is a glaring formatting error pointing towards to intentional manipulation to point at a secret author hidden within the initials of other authors.

The years and months the papers were published are unlikely to provide any additional information as to the identity of the author as they have not been chosen by the author. This leaves little opportunity for the author to mutate the numbers into a secret message leaving us

with the conclusion that it is unlikely we should be looking for additional decoding information from them.

We found this by examining the word tape within double-quotation marks. The choice of quotation marks has come under scrutiny by members of the public. There aren't many documents in the world that have come under so much scrutiny and found so few errors that they have blog posts about the formatting choice of quotation marks. Indeed, it appears these tiny formatting errors in convention that Satoshi appears to have made are signals to highly skilled individuals on the look out to verify the papers author. This individual unfortunately missed every single one of the errors we have picked up on so far but pointed out something we haven't. This is largely because the author of this report is not familiar with LaTeX. I can only assume that if the author of this post had figured out that it is a steganographic signal he would have either kept this little bit of information to himself or published his findings.

‘5. List of 6 steps

So far, we have found indications of steganographic manipulation in sections 7 and 10, our key is 7,2,5 someone versed in even the basics of computing which this paper is aimed at knows that 10 is binary for 2. So logically we should find some more steganographic manipulation in some section 5 completing the key we initially found.

Section 5: Network contains the documents most interesting feature from a steganalysis point of view, an alphanumeric list. The list 6 items long and the content around it don’t at first glance seem to be connected our list of numbers from the key. There are no noticeable “blemishes” or errors that we can see. A 6-item list is most likely to contain 6-character positions for 1 or 26 letters and possibly punctuation. While likely not enough to write out a full first, middle initial and last name including spaces, it may be enough to give us an initial and a last name or something similar.

Further, 5 being the last number in our key we might assume that this is the last decode section and thus the main puzzle which we believe we may have found the checksum for.

The list itself has some very interesting features, its usage of duplicate phrases while common in lists creates a visually interesting pattern. The most prominent duplication is at the start of some of the line “Each node” and “Nodes” separated by “When a node”. From a visual standpoint it is likely an important thing to note the lines get gradually longer for each point. While this looks satisfying it could also be being used to signal something related to our puzzle.

```
---

The steps to run the network are as follows:
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the
chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on
extending it. If two nodes broadcast different versions of the next block simultaneously,
some nodes may receive one or the other first. In that case, they work on the first one
they received, but save the other branch in case it becomes longer. The tie will be broken
when the next proofof-work is found and one branch becomes longer; the nodes that were
working on the other branch will then switch to the longer one.

---
```

The paragraph below the list mentions that a tie will be broken when one chain becomes longer. This could mean that we need to pay attention to the length of the lines throughout our decoding process to unpack the manipulation that has been done.

There are other interesting visual points on this page outside of section 5, including many hyphens and inconsistent spacings. For example, the sentence, “Proof-of-work is essentially one-CPU-one-vote.” seems to be surrounded by unnaturally large whitespaces. This sentence is one of only two in the entire paper to be unbroken by a line break. Although we are unlikely to be

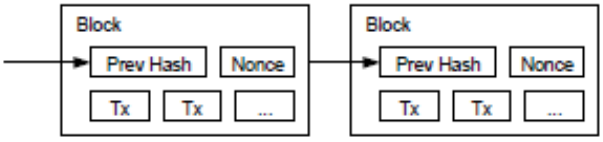
able to infer any instructions from this feature is it interesting to note as it potentially has uses to manipulate the positioning of other elements on the page.

To ensure the integrity of the elements positioning is preserved in this report below is a screen shot of the page:

4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

3

-- (Slight side bar)

Initially I didn't think much of something when doing this report but feel it is worth mentioning now while discussing the layout of the page. When copy pasting this section which, I've done it many times, I have noted a strange behaviour. Whitespaces are retained differently depending on what software you are copying from and pasting too. For this paragraph if we copy and paste this paragraph into notepad, on windows, using any browser but Microsoft Edge we get the below result:

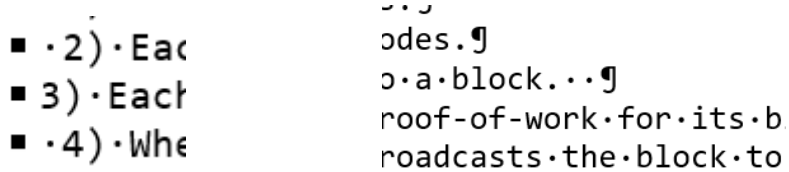
```
---
5.
 Network
The steps to run the network are as follows:
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the
chain, using the hash of the accepted block as the previous hash.
Nodes always consider the longest chain to be the correct one and will keep working on
extending it. If two nodes broadcast different versions of the next block simultaneously,
some
nodes may receive one or the other first. In that case, they work on the first one they
received,
but save the other branch in case it becomes longer. The tie will be broken when the next
proofof-work is found and one branch becomes longer; the nodes that were working on the
other
branch will then switch to the longer one.
---
```

With Microsoft Edge we get the below:

```
---
5.
 Network
 The steps to run the network are as follows:
 1) New transactions are broadcast to all nodes.
 2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
 4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
 5) Nodes accept the block only if all transactions in it are valid and not already spent.
 6) Nodes express their acceptance of the block by working on creating the next block in
the
chain, using the hash of the accepted block as the previous hash.
 Nodes always consider the longest chain to be the correct one and will keep working on
extending it.  If two nodes broadcast different versions of the next block simultaneously,
some
nodes may receive one or the other first.  In that case, they work on the first one they
received,
but save the other branch in case it becomes longer.  The tie will be broken when the next
proof
of-work is found and one branch becomes longer; the nodes that were working on the other
branch will then switch to the longer one.
---
```

There are two main differences, the first is that Edge interprets the first hyphen at the end of a line for "proof-of-work" as line break, but chrome and others do not, and **neither have the hyphen**. What is more, line item 3 is missing a space at the start causing it to be the only line not to have an indent. This is very peculiar behaviour. It seems to be caused by the line breaking

after a whitespace character instead of before it. This misplaces in line break maybe a result of manipulation to signal something.



This line 3 indent was first noticed by the author some time ago. I publicly pointed to this as a potential sign of steganography on February 9th, 2024, after the Dr. Craig Wright on the stand in a U.K. High court claimed that he had employed steganography as a method while authoring the Bitcoin White Paper. Although at the time and even throughout writing most of this report the purpose of it was not yet known as they had not taken the time to sit down and figure out what should have been an obvious method of claiming authorship.

From here we will assume that the length of a line is dictated by the number of characters it contains without whitespaces, although we will note this packet loss as we continue our decode. To help us read the paragraph properly we will clean them of data we are no-longer considering, yet still there, up until we no longer need them:

---

5. Network

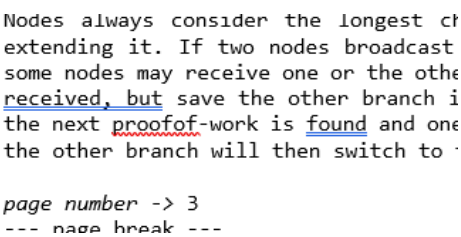
The steps to run the network are as follows:
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

page number -> 3
--- page break ---

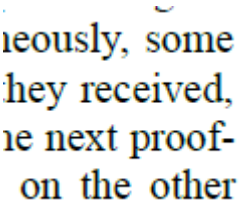
New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

Two mistakes have been picked up by Microsoft words spell check function while cleaning the data, so it is readable but with the same visible characters and the assumed reading. The comma after “received” in the paragraph below the list


Nodes always consider the longest ch
extending it. If two nodes broadcast
some nodes may receive one or the othe
received, but save the other branch i
the next proofof-work is found and one
the other branch will then switch to

page number -> 3
--- page break ---

shouldn't be there, and after the word "found" in the line below it suggests that there should be one. If the author had access to a spell check, we could speculate that they would have been informed of this grammatical error and corrected it. Again, this is a discussion of convention, but these kinds of conventions are what we have been considering as signals and thus that might be relevant.



On this line the commas position is right at the end of the line suggesting that it could simply be misplaced. Natural errors are much more common at the end of a line or page since they are not as visually obstructive in comparison to the emptiness of the negative space. Worth bearing in mind though as the missing comma after the word found is more in the middle of the line. If I correct these mistakes the spell check sees no errors at other than what could be overlooked as a software copy and pasting quirk in a missing hyphen.

Other features of this area include the documents only semi-colon (outside of the code 11. Calculations) and a duplicated phrase "becomes longer" on the same lines as the misplaced comma.

Highlighting duplicated words shows us some interesting patterns.

```
The steps to run the network are as follows:
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already
spent.
6) Nodes express their acceptance of the block by working on creating the next
block in the chain, using the hash of the accepted block as the previous hash.
```

Due to the nature of a what this list is detailing we would expect certain things to be repeated but the exact repetition and position seems deliberate and possibly indicating something. NEEWNN being the first letter of each line also has an interesting pattern to it, four unique letters like tape. Without knowing how it might be related to what we have already found we can't do anything with this potential information. It may suggest that we separate the list into parts, the unique W being the only nonduplicate lines up with our indentation when we copy and paste the list. We could speculate that it is in 2 parts, part 1 steps 1-3 and part 2 steps 4-6 but until we have more information about what we might be doing we can't know how this is affecting things.

The second paragraph is 4 sentences long which have been separated and compared below:

```
---
Nodes always consider the longest chain to be the correct one and will keep working on
extending it.

If two nodes broadcast different versions of the next block simultaneously, some nodes may
receive one or the other first.

In that case, they work on the first one they received, but save the other branch in case
it becomes longer.
```

```
The tie will be broken when the next proofof-work is found and one branch becomes longer;
the nodes that were working on the other branch will then switch to the longer one.
---
```

The first line has no punctuation but next 2 have a comma and the last has a semi colon. Since we are looking at punctuation to separate and compare lengths it is worth noting that this semi colon choice could also be important, although we aren't sure why yet.

Doing the same again for the second paragraph on the next page:

```
---
New transaction broadcasts do not necessarily need to reach all nodes.

As long as they reach many nodes, they will get into a block before long.

Block broadcasts are also tolerant of dropped messages.

If a node does not receive a block, it will request it when it receives the next block and
realizes it missed one.
---
```

This doesn't seem to produce anything of value. We could continue to do this, or we could reconsider what we are doing and looking for and what we have seen so far.

Performing these copy paste operations is frustrating, when copy and pasting the last sentence of the paragraph immediately under the list in words "proof-of-work" the first hyphen is always dropped. It doesn't matter which software we use it always pastes as "proof of-work" or "proofof-work" with a line break. It is almost as if this hyphen isn't a hyphen character in the paper, but a line placed there by the author and pieced together to indicate something.

Hyphenation Haven

There are many hyphens on this page and interestingly in one area in particular no hyphens where maybe there should be.

4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proofof-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

In section 4 Proof-of-Work word processing software conveniently indicates there should be a hyphen within “a number of zero bits”.

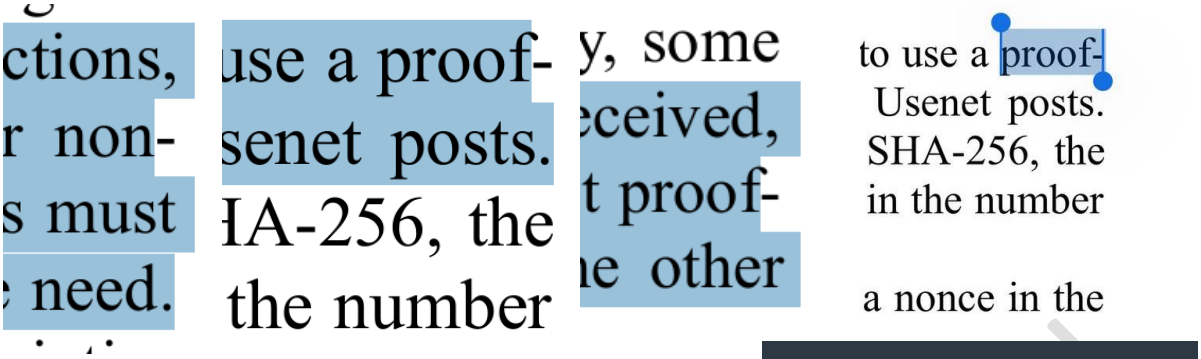
ves scanning for a value
ith a number of zero bits.
F zero bits required and ca

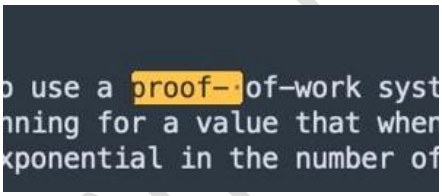
As we mentioned towards the start the document is covered in hyphens between various conjunctions below is a lite of all of them:

proof-of-work	<- 18 times	
proof of-work	**<- 2 times**	[no-hyphen/signal?]
peer-to-peer	<- 6 times	
double-spending	<- 4 times	
double spending	**<- 1 time**	[no-hyphen/signal?]
double-spent	<- 1 time	
non-reversable	<- 2 times	
non reversable	**<- 1 time**	[no-hyphen/signal?]
a number of zero bits	**<- 1 time**	[no-hyphen/signal?]
SHA-256	<- 1 time	
fan-out	<- 1 time	
multi-input	<- 1 time	
one-CPU-one-vote	<- 1 time	
one-IP-address-one-vote	<- 1 time	
mint based	<- 1 time	[spell check/signal/style?]

Starting with the last on the list, mint based. Today and in this paper, this would seem to be an obvious place to put a hyphen if we are looking at hyphens. We haven’t found

Two are clearly errors that could be interpreted as a signal due to their proximity to our steg decode and references. Also due to the form of how the error is presenting in that: there is in fact a line on the page but not in a form that allows it to be highlighted using the cursor thus can’t be copy and pasted. Both instances in the proof of work case are on the same page in the first couple of line at the top of the page and bottom of the page. These could have been used are targets for a steganographer to try and line up as they are adjusting the spacing of the document.

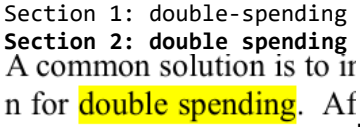




Noted by collegue, contributer and friend Jerry David Chan, on a macintosh using safari and on iphone using chrome he is able to highlight the hyphen. Further when he copy and pastes it into notepad he is able to notice an additional space but no line break, this looks to be a difference in software specifics and would be expected if there was any intnetional manipulation as there is no standard way to deal with an author intentionally inserting what you might think of as error into their own document.

The “double spending” instance on page 2 section 2 doesn’t have a hyphen at all, every other time in the paper the author uses “double-spending”, the occurrences in order are.

```
Abstract:  double-spending
Abstract:  double-spending
Section 1: double-spending
Section 2: double spending <- missing hyphen
```



I used the number 2 as a reference which is where and why we speculated intentional steganographic manipulation in the first place. This missing hyphen in the word “double-spending” is dubious as it is inconsistent. It could be seen as an instruction that we are not looking in the right place, but we are looking for the right kind of thing. Instead, the secret writer is guiding us and wants us to make the logical leap of faith that in a paper about computer networks the number 2 might also be represented in binary form as the number 10 which is where we found “tape” leading us to the CheckSum.

Now that we know the hypothesis that hyphens are being used as an element to signal instructions just got a little firmer, we can examine the missing hyphens more closely.

The hyphen missing in “non reversable” is also at the end of a line and maybe due to the way the software is working but we can’t know if it is intentional or not without further examination of why it might have been used.

In the sentence itself it reads:

```
The cost of mediation increases transaction costs, limiting the minimum practical
transaction size and cutting off the possibility for small casual transactions, and there
is a broader cost in the loss of ability to make non-reversible payments for non reversible
services. With the possibility of reversal, the need for trust spreads.        ^-(missing)
```

If the author is using hyphens to signal something this could be the author signalling very subtly that they are aware that in releasing this paper and software there is no going back, and nothing can stop the world changing from that point forward or simply an error that the author didn't intend due to it being at the end of a line. We will bear it in mind as we continue our analysis.

Strangely though both instances for "proofof-work" occur on the same page, top and bottom. Almost as though they might be markers for formatting and may be related to one another. No other page has a characteristic like this. They could be signalling something much more specific, a possible sign of steganographic injection via space formatting is a common method after all. To understand that we need to look more closely at the 2nd instance at the top of the page that we haven't closely examined yet. As such this marks a fresh area of interest. Signals normally must be near one another or in logically defined places depending on the layout of the puzzle.

The instance at the top of page 4 is in the same sentence as Reference [6] and Adam Backs name. The next time hyphens should be used, but are not, is towards the end of the paragraph with short phrase "number of zero bits." should have at least one hyphen in it between 'number' and 'of' or changed to "several zero bits". To correct this one would normally insert them and move on. However, if used as a signal as a task for us to complete like we are doing with the rest of our manipulations then the instruction a secret message like this is what you would expect at first glance conceptually. Let us examine it further:

When reading and making edits to a document we typically work in a process of reading, considering, demarking, addition, moving on. While the editing process is all these things, an error of this type is a process interruption, leading us to believe that this might be an edit that was unfinished due to an author getting distracted with something else. However, if we are to assume that there is a secret message to be decoded then, this can be viewed as an instruction to finish a process that is still open.

Markup code for striking out a character is typically in the form of:

```
Text ~~text to remove~~ text -> Text ~~text to remove~~ text -> Text text
```

If we are using a steganographic method to hide this information using a double Tilde (~) or even one might be too obvious for even a casual observer that this is clearly a typo. So, we hide this face by making the conceptual element smaller as a hyphen. In the end literally straightening out the unevenness in the document.

As a process when reading the English language it is expected that an editor would conventionally move with the direction of reading, thus process output of each step in the process would look like the steps below and an interrupted process would get stuck at step 2.

```
Text that and editor likes (text the editor doesn't like) continuation of the text the
editor likes.

Text that and editor likes [-] (text that is marked as removed) [ ] continuation of the
text the editor likes.

Text that and editor likes [-] (~~text that is marked as removed~~) [-] \things that should be
"inserted instead" / continuation of the text the editor likes.
```

```
Text that and editor likes [\] things that should be “inserted instead” [/] continuation of
the text the editor likes.
```

writer the author is providing instructions for us to finished would traditionally be infer striking off (strike-through) the section. If we do so Adam Back’s name and his reference is crossed out.

```
To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a
proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet
posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-
256, the hash begins with a number of zero bits.  The average work required is exponential
in the number of zero bits required and can be verified by executing a single hash.
```

We can easily infer from this that the author is trying to subtly communicate that they are not in fact Adam Back. In steganography terms this is a very big slap in the face, the person encoding the message communicating that they wish to disassociate themselves from this individual and his work. Something personal.

If you ever construct your own puzzle, it may feel like the cover and content start to become merged in such a way that it is extremely difficult to decern what is and is not part of the set of steganographic instructions until you learn all the rules.

There may be additional information as to which specific words are to be struck through. Let’s quickly investigate this possibility.

The first missing hyphen when copying the text is the first on in proof-of-work, then we have 4 before the position we should have one in the phrase “number of zero bits.”

```
      <-- Start ---------------------- End -->
     V 1           2  3         4          V
proofof-work, proof-of-work, SHA-256, number of zero bits.
```

Where there are 3 potential positions we could choose from and one choice requiring 2. Yet, the choice of either would mean that we might not notice a connection between a missing one at the start of the paragraph and the one at the end. 4 hyphens may infer that we skip the first 4 words and then select the next three words to cross out producing a much more precise version:

```
To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a
proofof-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet
posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-
256, the hash begins with a number of zero bits.  The average work required is exponential
in the number of zero bits required and can be verified by executing a single hash.
```

I elected to remove [6] as a 4th element as well because the removal of him and his invention from the paper would also negate the reference. As the amount that is removed is a little ambiguous from my standpoint. I feel Steganographer, would want the sentence to retain logic after the point is removed, leaving the other option that everything is removed which doesn’t really improve the problem that much due to the resulting “a proof bits”. Sometimes something is good enough to be considered complete.

```
To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a
proofof-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet
posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-
256, the hash begins with a number of zero bits.  The average work required is exponential
in the number of zero bits required and can be verified by executing a single hash.
```

The skill required to so subtle align the hyphens with the line breaks such that no one would notice while seems inconceivable on its surface isn't completely out of the realm of possibility. Doing so would only require the following steps not matter how sophisticated the software:

- Write sentance with hypens in a normal way.
- Restructure elements so they are all roughly in the right place
- Adjust settings that affect the position of the hyphens such as:
    - Font size, spacing
    - Word spacing
    - Line spacing
    - Paragraph spacing
    - Diagram size/padding
    - List padding
    - Other things
        - This should be enough
- Get all elements are on the same page
- Adjust settings until the hyphens are at the end of the line
- Convert the hypen to vector or otherwise replace it with one
- Save document.

Adam and Satoshi

Given that Adam back received the first email from Satoshi in 2008 and it was later made public in a rather notable court case concerning the identity of Satoshi Nakamoto with Craig S Wright being on the other side. It appears Adam Back and Satoshi may have had some kind of personal disagreement from before the Satoshi Identity was created. Adam Backs Hashcash paper, reference [6], is noted as non-original work, in the introduction it states:

```
At the time of publication of [1] the author was not aware of the prior work by Dwork and
Naor in [2] who proposed a CPU pricing function for the application of combatting junk
email.
```

Satoshi being a so meticulous in his effort to produce such a paper and technology and then embedding that paper with secret messages using the ancient art of steganography would probably have preferred to cite original work where possible. The list of references included go back as far as 1957, Dwork and Naors work on pricing via processing was published in 1992, just 5 years before Adams work. This truly original paper is going to be something that Satoshi looked at and considered as a reference. Instead, he decided to give Adam Back the option of correctly attributing the invention of proof-of-work.

In his first publicly recorded email Satoshi writes to Adam:

```
---
From:   "satoshi@anonymousspeech.com" <satoshi@anonymousspeech.com>
Sent:   Wed 8/20/2008 6:30:39 PM (UTC+01:00)
To:     adam@cypherspace.org
Subject: Citation of your Hashcash paper
---

I'm getting ready to release a paper that references your Hashcash paper and I wanted to
make sure I have the citation right. Here's what I have:

[5] A. Back, "Hashcash a denial of service counter-measure,"
http://www.hashcash.org/papers/hashcash.pdf, 2002.

I think you would find it interesting, since it finds a new use for hash-based proof-of-
work as a way to make e-cash work. You can download a pre-release draft at
http://www.upload.ae/file/6157/ecash- pdf.html Feel free to forward it to anyone else you
think would be interested. I'm also nearly finished with a C++ implementation to release as
open source.

Title: Electronic Cash Without a Trusted Third PartyAbstract: A purely peer-to-peer version
of electronic cash would allow online payments to be sent directly from one party to
another without the burdens of going through a financial institution. Digital signatures
offer part of the solution, but the main benefits are lost if a trusted party is still
required to prevent double-spending. We propose a solution to the double- spending problem
using a peer-to-peer network. The network timestamps transactions by hashing them into an
ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without
redoing the proof-of-work. The longest chain not only serves as proof of the sequence of
events witnessed, but proof that it came from the largest pool of CPU power. As long as
honest nodes control the most CPU power on the network, they can generate the longest chain
and outpace any attackers. The network itself requires minimal structure. Messages are
broadcasted on a best effort basis, and nodes can leave and rejoin the network at will,
accepting the longest proof-of-work chain as proof of what happened while they were gone.
```

To which Adam Adam responded:

```
"Yes citation looks fine. I'll take a look at your paper. You may be aware of the "B-money"
proposal, I guess google can find it for you, by Wei Dai which sounds to be somewhat
related to your paper. (The b-money idea is just described concisely on his web page, he
didn't write up a paper)."
```

Whois Record for Upload.ae

— Domain Profile

Registrar	Etisalat IANA ID: – URL: – Whois Server: –
Registrar Status	ok
Name Servers	NS31.MACBERRY-HOST.COM (has 193 domains) NS32.MACBERRY-HOST.COM (has 193 domains)
IP Address	139.64.178.20 - 160 other sites hosted on this server
IP Location	- New Jersey - Hackensack - Whitelabel It Solutions Corp
ASN	AS394625 WHITELABELIT, US (registered Nov 02, 2015)
Hosting History	16 changes on 4 unique name servers over 10 years

Whois Record (last updated on 20250302)

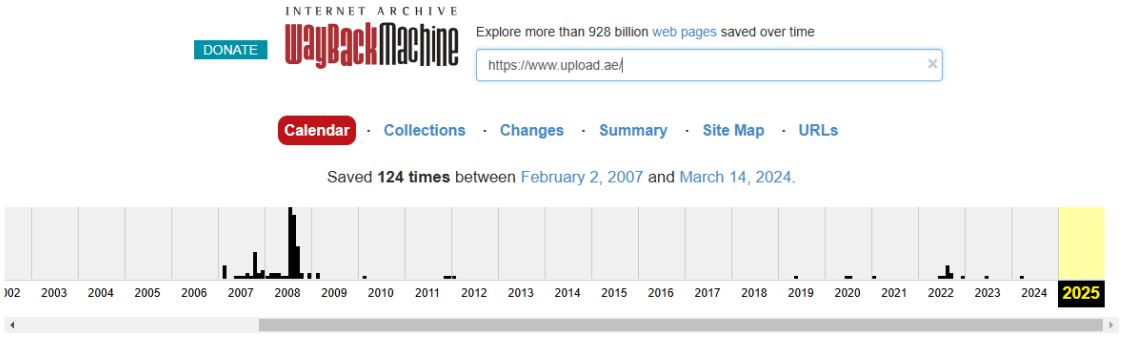
```
Domain Name:                    upload.ae
Registrar ID:                   Etisalat
Registrar Name:                 Etisalat
Status:                         ok

Registrant Contact ID:          ETSLT453370-R
Registrant Contact Name:        Mohammed Alsayed
Registrant Contact Email:       Visit whois.aeda.net.ae for Web based WhoIs
Registrant Contact Organisation: Mohammed Alsayed

Tech Contact ID:                ETSLT453371-C
Tech Contact Name:              Mohammed Alsayed
Tech Contact Email:             Visit whois.aeda.net.ae for Web based WhoIs
Tech Contact Organisation:      Mohammed Alsayed

Name Server:                    ns31.macberry-host.com
Name Server:                    ns32.macberry-host.com
```

While short this exchange says a lot more than you might think. For example, the link was hosted by upload.ae. which doesn't appear to have a web archive link for this document although people attempted to record something in 2020 when these emails were first made public. This is an interesting choice to deliver the file to Adam, not only because it is niche but because a PDF can be attached to an email no problem. Uploading it to a server of a particular service may also tell us something about the relationship that Satoshi has with Adam Back. The hosting provider seems to have suspended the account after the website was sold some time in 2020 according to their web archive captures. When the site first launched, they had a flurry of web archive links created around 2008, they appear to have had their first crawl at the start of 2007 and registered the domain at an unknown time as the whois information only shows the domain being registered at some point before 2015 it became unregistered.



As Adam is a CypherPunk and claims Satoshi was if we examine the CypherPunks mailing list for we might find an announcement of this domain.

Although not in plain text, an advert is found by searching for the world upload on the cypherpunks archive of marc.info. Going through the emails around this time looking for potential uses of steganography we can immediately find something interesting.

The email is posted below:

```
---
List:       cypherpunks
Subject:    What up, ,Come see me again.
From:       "Carlo Brandon" <mareesa () mobyfan ! com>
```

```
Date:       2005-04-02 14:03:37
Message-ID: 20040512191020.56043.qmail () web21501 ! mail ! yahoo ! com
---

"You've got to check out my new site; Somebody taught
me how to upload them last nite; Im pretty new at this but
check em out and see what u think!"

Pics are HERE!
Copy and pa ste the addre.ss below and put it into your browser.
Http://www.dioderidden.aerosol.pettlespuzzle.com/cs2/




dont want no more
httP://www.intrepid.southsideplaya.com/retract/

throwaway acreage silk phosphorescent cameramen drizzly yankton chloride. harcourt \
sepulchral maier newell. potassium card.
---
```

We can see plainly that the plaintext written URL in the email body is not valid. Also, the subject line contains a substantial error with 2 commas randomly inserted into the middle of a nonsense and uninformative subject line.

Performing a very quick steganalysis:

The errors within the quotation marks within the body of the email the missing apostrophe from “Im” and the first one being immediately preceded by the letter u suggests that we are looking for a word starting with ‘u’ which would be upload.

This is subtly confirmed by the usage of ‘u’ as opposed to ‘you’ in the second half of the sentence.

As for the TDL (Top-Domain-Level) we can find it above the plain-text domain they give us. We can confirm this in a similar manner to confirming the upload extraction by noticing the emphasis on the letter e in the second half of the quotation and the capitalized HERE above the “pa ste” error. Suggesting ‘ae’ as a TDL giving us a result of: “upload.ae”

The sender of this email has only ever sent this email to the mailing list suggesting that it is a place for nefarious activities that people would rather not be associated with.

Maybe the sender provided a particular file that the user uploaded if we were to continue to decode it. I expect you as a regular citizen of the world are as uninterested in the contents of any such secretly advertised URL as I am. When these are the methods used to find your customers there aren’t many places normal people will want to venture. We have the information we need to form logical assumptions about the people involved and move forward with our analysis.

We can only assume this is a secretive advertisement for the file sharing service Satoshi used, thus we can infer that not only did Satoshi have knowledge of this kind of steganographic method if he was a real cypherpunk but that Adam Back as a cypherpunk and avid participant of the Steganography mailing list also is.

Although I have not formally decoded this email, I'm not sure it requires it since even to the casual observer there are some quite drastic errors in it. To demonstrate a different kind of puzzle, Rand al'Thor posted to puzzle.stackexchange.com an email that needs to be decoded. Yes, people really do this for fun as well. This puzzle doesn't seem to use TFEs as we have, but maybe you will be able to solve it:

You are a secret agent in the service of the KGB, about to embark on a highly dangerous mission to infiltrate MI6. You have your disguise, your papers, and your backstory all prepared. The night before your departure, you receive the following email:

```
From: Andrew Void < a.void@disparition.com >
Sent: Fri, 27 Mar 2015 11:57AM +0400
To: ████████████████████████
Subject: Your work

Dear Mr Smith,

This is to inform you that your poem is now nearly noted up for publication. Its age
notwithstanding, this poem will fit as part of a vast pattern of poems that spans
millennia. You stand among us now as a poet, throned among such applauded poets as
Aristophanes, Plato, Byron, and so on. As one of us, your poetic prowess will not go
unadmired.

Many congratulations!

Andrew
```

Mr Smith is the pseudonym you will be adopting on your mission in Britain, but you do not recognise the name Andrew Void or the email address. You are about to delete the email as spam, but some instinct tells you to examine it more closely. After a few minutes at your computer, you find the hidden message within it and slump back in your chair, disappointed.

What is the hidden message?

There is rather large hint further into this document since knowing that piece of information is quite important for your ability to decode this one.

When compiling this document in its final form and reviewing the overview of each section it occurred to me. When first created the overview list for this analysis I noted that there are no visual anomalies, it is the most basic section with no diagram, references, hyphens or anything of that nature, only two instances of CPU capitalised. Reference [6] is also Adam Backs reference number, the fact that it is baren except for these capitalisations my indicate that Craig Wright as author of the Bitcoin White Paper believes Adam has contributed little if anything of value.

The positioning of CPU surrounding a paragraph relating to transaction fees providing the incentive and no longer requiring new coins to enter circulation providing a system that is inflation free may say something. Adam Back et al. due to either their inability or unwillingness to scale the number of transactions that BTC can process, have been discussing the possibility of removing the limit on new coins meaning that whatever system they may oversee will not conform to this paragraph of the paper.

It is quite well established in the Bitcoin zeitgeist that the number of coins is fixed, at least it was at the start. It isn't clear what other "features" and "capabilities" may be added, but their plan is to "ossify" at some point

.

Hashing it out, – in public

Returning to our White Paper Steg puzzle from section 5 and the last in our key. Typically, in steganography puzzles the solution is related to the context in which this puzzle is found. This is a technical paper describing a network process to automatically economically determine the winner in a GAME-OVER successive rounds. In this puzzle are searching for some unique string that has meaning related to the game this author is playing. Maybe the final message isn't a name at all but a taunt to antagonise certain people the invention threatens just that little bit more. The invention is entirely based on game theory. It is possible that Satoshi used this area to simulate how the winners and losers within his invention are determined.

We may be able to estimate how the list is supposed to work by understanding the network. This needs to be determined somehow. Our overall goal is to have each line or elements with the line select a letter conforming to some basic rules that we might be able to interpret from what we are seeing. This is typically what steganalysis entails.

Using the theory that a letter would be representative of a hash string, or "a number of zero" bits, then our competitors in each round of the game must produce a certain string for the winner to be determined. The winners being the word or letter that occurs the most. This is frequently otherwise known as frequency analysis. It is a standard tool in a steganographers tool kit. While we may not be exactly right about how we think this will work, experimenting will help us understand if we are on the right path.

If correct, then we can probably expect a starting state to be denoted somehow, maybe with possible contenders and then reach round separating the letter we are looking for in each instance. The starting state would logically be towards the top of the list. We don't know how many rounds we are searching for but at this early stage lets assume that a round is a numbered step in our process outlined in the list giving us a message of 6 characters.

To see more structure in the steps and considering a block in bitcoin is identified uniquely with "a number of zero" bits at the start we can assume that we should perform a unique word frequency analysis over the text that start in a similar way. For example, accept, acceptance, accepted. To do this I have simply coloured each unique word that appears more than once.

Quickly checking for duplicated words in each line:

```
The steps to run the network are as follows:
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already
spent.
6) Nodes express their acceptance of the block by working on creating the next
block in the chain, using the hash of the accepted block as the previous hash.
```

This doesn't seem to give us much indication of anything other than as a more visible overview of how evenly each word and thus character is distributed throughout the text.

Starting simplistically at the first line and ordering by the starting letter frequency we get:

```
1) (New, nodes), (Transaction, to), (are, all), (broadcast)
```

Revealing a first letter frequency of:

```
(N,2), (T,2),  (A,2), (B,1)
```

When performing this first search, we immediately encounter a problem. We have three contenders for our letter. When considering words at this level, having a number of letters at equal positions is very likely going to be the case in every line we try to decode. We could analyse the frequency of letters but are likely to encounter a similar problem in that there are a few very common letters that will appear much more frequently than others. Creating noise in our result. The most common example being the letter 'e'.

The letter 'e' is so common among Latin alphabet languages that French author Georges Perec proved his mastery of language by writing an entire novel without a single instance of it. This technique of adding additional limits to the language you use in a piece of text or writing is called Oulipo (pronounced Ou-lip-o) constraints. Oulipo is an abbreviation of Ouvroir de littérature potentielle; (workshop of potential literature). A similar technique to what we are currently undertaking. Interestingly from our perspective the book titled "La Disparition" and published in 1969, translated into English by Gilbert Adair under the title "A Void" in 1995 a disappearance is a central theme and point of the book. It has been translated into many different languages of various skills, some of whom have successfully managed to keep the feature of not including 'e'. Obviously, some languages such don't lend themselves well to this restriction, but that is the entire point. To prove something to either the world or; the author wishing to prove something to themselves.

To reduce noise the author could have inferred several instructions that add rules to our current assumed instructions. Since we are facing a granularity problem, we are likely looking at a battle of character frequency as opposed to words. The problem that would need to be solved by the encoder and decoder is that of noise generated by useless letters.

Before looking for the other errors we noted as potential instructions it may help to re-examine our list so that we can understand what kind of instructions we might be looking for:

```
1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already
spent.
6) Nodes express their acceptance of the block by working on creating the next
block in the chain, using the hash of the accepted block as the previous hash.
```

Notes on distribution of words and thus letter frequency.

- "(New) Transactions" and "broadcast(s)" are found towards the top
- Node is quite uniform throughout
- Proof of work is hyphenated and in the middle of the text lines and columns.
- Block is more heavily waited to the last points with line 6 containing 3 instances.
- Hash is mentioned twice in the last step.

Now examining the text below it as two paragraphs:

```
Nodes always consider the longest chain to be the correct one and will keep working
on extending it. If two nodes broadcast different versions of the next block
simultaneously, some nodes may receive one or the other first. In that case, they
work on the first one they received, but save the other branch in case it becomes
longer. The tie will be broken when the next proofof-work is found and one branch
becomes longer; the nodes that were working on the other branch will then switch to
the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as
they reach many nodes, they will get into a block before long. Block broadcasts are
also tolerant of dropped messages. If a node does not receive a block, it will
request it when it receives the next block and realizes it missed one.
```

Recalling that we have removed the line breaks which do have an error within them but we are for now at least focused on what we can see, we can see the missing hyphen which the misplaced line breaks seem to have caused, these are highlighted in red.

Highlighted in blue is a missing comma indicated by spell check, these mistakes at in very close proximity and maybe related. Also, in proximity are the other features we noted, the documents only semi-colon and a comma that if moved from its current position after received and input after found fixes the spellcheckers error. We can't know if we are using the same spellcheck that the author used but in today's modern world, they all have a standard ruleset for each language and subset thereof.

Given that we have punctuation out of order in the paragraph below the list we can probably assume that this demarks the starting point or ending points of the strings, we must analyse but also that we may have to reorder them. It maybe easier to examine the punctuation in each while comparing their position more thoroughly. The list its punctuation and the associate errors from the paragraph below to are separated and analysed below, we have included the top line above the list this time due to its use of colon which maybe important as we are examining this type of punctuation mark. We have also replaced each number with its index in the line starting from 1.

```
The steps to run the network are as follows:
1) [1] [2] [3] [4] [5] [6] [7].                ^-- Colon

2) [1] [2] [3] [4] [5] [6] [7] [8].

3) [1] [2] [3] [4] [5] [6] [7] [8],
   [1] [2] [3].

4) [1] [2] [3] [4] [5] [6], [1] [2] [3] [4] [5] [6] [7].

5) [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16].

6) [1] [2] [3] [4] [5] [6] [7] [8] [0] [10] [11] [12] [13] [14] [15] [16] [17],
   [18][19][20][21][22][23][24][25][26][27][28].
```

We likely have events, message or branches denoted by commas or periods. If this is the case one would logically think that we would need one at the start of the game. It is of note that each of our spaces is a letter and that there is plenty of room within each line for a steganographer to play a matching game with themselves. Running a little piece of software for each revision, then

copy and pasting it from the final document running the same analysis they would be confident that their method would be forever enshrined within the words they have written. We are hoping piece together this same analysis resulting in a message they embedded. For a Steganography proof it is a little fuzzier than that. The principles of Steganography would dictate that the three-letter string we are assuming are the initials of the author must be related to the information we obtain. If it is not, then we aren't finished. The message we have already extracted should be used as a check sum of the message. Although the proof is in the method, and cover, it is not running some random piece of software over random paragraphs with random parameters to hopefully generate something of meaning. There must be a simple way within the document to verify the message extracted so that any reader that wishes to verify the message with its sender they could see if the message extracted matches the shorter abbreviation.

Listing the paragraphs below the list and separating them by their every punctuation mark and ignoring the letters we get the following:

```
1       [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] [].

2       [] [] [] [] [] [] [] [] [] [] [],
3       [] [] [] [] [] [] [] [] [].

4       [] [] [],
5       [] [] [] [] [] [] [] [],
6       [] [] [] [] [] [] [] [] [] [].

7       [] [] [] [] [] [] [] [] [] [] [] [] [] [] [] []; <-- Semi-Colon

8       [] [] [] [] [] [] [] [] [] [] [] [] [] [] [].
```

This pattern is:

```
[ . , . , , . . , .]        <- 9 elements
```

This doesn't seem to match the pattern from our list which is:

```
[ . . . . . , . . . , .]    <- 11 elements
```

Although discounting the colon we reintegrated we get the same number of elements they don't seem to form a consistent pattern as the number of commas within each.

```
[ . . . , . . . , .]        <- 9 elements (without colon)
```

There are two extra commas and 2 less periods, this may indicate which lines need to be altered to get a readable message, but we can't infer how they should be manipulated.

After quite a bit of deliberation, on whether or not certain things are actually errors or not, I have come to the conclusion that if these errors are indeed intended to be communicated via incredibly subtle slight of speech and an assumption that this document would be in detail like we are then the spell check of common word processing software would be a powerful tool utilise to inject errors into anyone's analysis. Lining up the hyphenations with the ends of the line is a very subtle way of injecting steganography into a steganalysis that some may perform. This works particularly well in cases such as this where there is only one known format that the publisher and file that we know was used and published by the author. The person encoding

these messages must have some form of social issue as it is an incredibly carefully considered error to make injected in such a way that it sounds preposterous to anyone that has not been through the decode step by step. Spellcheckers, however, are a very useful tool to help indicate what might or might not be a manipulation that we are to interpret as instructions to an eagle-eyed reader.

When looking at the sentences below our list:

```
Nodes always consider the longest chain to be the correct one and will keep working
on extending it.

If two nodes broadcast different versions of the next block simultaneously, some
nodes may receive one or the other first.

In that case, they work on the first one they received, but save the other branch
in case it becomes longer.

The tie will be broken when the next proofof-work is found and one branch becomes
longer; the nodes that were working on the other branch will then switch to the
longer one.

New transaction broadcasts do not necessarily need to reach all nodes.

As long as they reach many nodes, they will get into a block before long.

Block broadcasts are also tolerant of dropped messages.

If a node does not receive a block, it will request it when it receives the next
block and realizes it missed one.
```

Most normal people have learned to largely ignore spellcheck errors if they feel the software is being too fussy about what we are trying to write. When I noticed the following error, it was picked up by the spell-checking functionality indicating an unnatural choice of words and that we should reconsider our phrasing. This made me rethink the sentence this error is within. It is in fact, worded very clearly from a steganographic perspective.

```
As long as they reach many nodes, they will get into a block before long.
```

The spellcheck functionality suggests “As long as” should be “IF.” The fact that this sentence ends with the word spellcheck is suggesting we remove. This sentence behaviour is common in instructions and seems to confirm this suspicion. We should also note that the word before long at the end of the sentence is before. We can assume that this will either mean we end our removal of letters before the end of the word “long”, or before long has finished leaving us with a ‘G’.

```
As long as they reach many nodes, they will get into a block before long.
      IF
```

The most complex version of this instruction that we could consider is that we should remove all instances of ‘a, s, l, o, n’ and replace I with F or F with I. It may be a bit of a stretch to know that a spellcheck suggestion could be used so precisely so if we are swapping letter, we should look for a signal that might be clearer than this. It is unlikely that the author is inferring that we replace all or some of the letters with either I or F or some kind of combination due to the dead-

end this investigation would hit and the number of possible combinations this creates being too numerous for anyone to reasonably decode without brute forcing an answer out of the text and ambiguous rules.

It seems the concept we are dealing with is that of a “game to produce letters”, a blockchain where each round is denoted by punctuation. The highest frequency letter wins the race. We should note that we will have packet loss due to our method of reducing noise. The method we found to reduce the noise with a slight modification to our understanding could also be used to amplify a signal. We can look at the other errors in the paragraphs below again:

```
In that case, they work on the first one they received, but save the other branch in case
it becomes longer. The tie will be broken when the next proof-of-work is found and one
branch becomes longer; the nodes that were working on the other branch will then switch to
the longer one.
```

As we mentioned the spellchecker is advising that we move the comma from after received to after found. Received and found are things that our competing letters are doing, sometimes receiving letters others have found and sometimes finding the letters themselves. If we are to take these instructions at face value, we might be expected to replace the letter F with R at some point in our analysis.

On the topic of swapping things, it is common in steganography when manipulating things to only perform one action at a time, this if we are going to need to swap lines around during the analysis then these lines will be next to each other unless we find specific instructions.

We haven’t seen an instruction containing the letter e which we originally pointed out is a source of a large amount of noise when performing such analysis, this infers that it is either a consideration we are assumed to undertake as we are already clearing noise, or that one of our letters is E.

Finally, there is something missing which order do we work in and with which line do we start?

Reconsidering the other errors we have come across the missing hyphen within “non reversible” may be relevant and not just one overlooked when dealing with so many across the document in such an instrumental way. Seems unlikely to be a coincidence at this point that one of the instances of “non reversable “is either missing an e for the word “none” or missing a hyphen. We could consider that this is a signal to also drop the “e” like other letters but also that if we run the simulation in reverse, starting at the last period we should find our answer.

Finally, before we move onto experimenting with our letter frequency analysis over the 6-step list in the Network Section of the Bitcoin White Paper, lets summaries the instructions we think we have so far and where they come from:

- Count number of letters in line to get winner
  - Inferred by Bitcoin network mechanics
- Start from last point at last letter
  - Missing hyphenation in non-reversible
- Noise Reduction
  - Remove letters AS LON & E
    - “As long as, before long”

        - "NonE reversible" steganographicly valid yet, not grammatically.
    - Signal Amplification
        - Replace F with R
            - Comma moving from received to found.
    - Change order of lines
        - Comma moving
        - Semi colon
        - Colon
    - Expect packet loss
        - Inferred by Bitcoin network mechanics and steganography process
        - Plain text.

This is a lot of manipulation indicating the difficulty in producing a legible answer.

Traditionally people expect that doing this type of decoding work takes understanding complex pieces of software with so much customization abilities that we would essentially be trying to brute force every permeation of the cover text possible. However, this is never how steganography in practice works for text applications. Sometime the author cares so much about their work that they wish to secretly communicate the works author within the text itself. This would be especially common if the author wishes to use a pseudonym at publication. Often the only way to be assured your work is correctly preserved and correctly attributed no matter what else happens is to embed your name within the words themselves. To analyse this, we need little more than a spreadsheet to help us in our counting.

Pasting our strings into a spreadsheet and using the formula:

=LOWER(REGEXREPLACE([string],[select], [replace]))

We can use columns to remove the characters one after the other so we can check the process as we go. First removing all the white spaces and leading unecassary characters while unnecessary will help you the audience see what is happening at each step and why. We have also added a new naming convention for each line in the order indicated by the missing hyphen in reversible.

# End STRING		misc.
9 :	thestepstorunthenetworkareasfollows	
8 .	Newtransactionsarebroadcasttoallnodes	
7 .	Eachnodecollectsnewtransactionsintoablock	
6 .	eachnodeworksonfindingadifficultproofofworkforitsblock	--
5 ,	whenanodefindsaproofofwork	--
4 .	itbroadcaststheblocktoallnodes	
3 .	nodesaccepttheblockonlyifalltransactionsinitarevalidandnotalreadyspent	
2 ,	nodesexpresstheiracceptanceoftheblockbyworkingoncreatingthenextblockinthechain	
1 .	usingthehashoftheacceptedblockasthepreviousHash	

Removing the letters: [e,a,s,l,o,n,g,]

```
# End STRING                                                                misc.
9 :   thtptruthtwrkrfw
8 .   wtrctirbrdcttd
7 .   chdcctwtrctiitbck
6 .   chdwrkfidididifficutprffwrkfritbck                                    --
5 ,   whdfidprffwrk                                                         --
4 .   itbrdctthbcktd
3 .   dccptthbckyiftrctiiitrviddtrdypt
2 ,   dxprthirccptcfthbckbywrkicrtithxtbckithchi
1 .   uithhhfthccptdbckthprviuhh
```

Our final manipulation replacing f with r we get the following:

```
# End STRING                                                                misc.
9 :   thtptruthtwrkrrw
8 .   wtrctirbrdcttd
7 .   chdcctwtrctiitbck
6 .   chdwrkridididirricutprrrwrkrritbck                                    --
5 ,   Whdridprrrwrk                                                         --
4 .   itbrdctthbcktd
3 .   dccptthbckyirtrctiiitrviddtrdypt
2 ,   dxprthirccptcrthbckbywrkicrtithxtbckithchi
1 .   uithhhrthccptdbckthprviuhh
```

Using these strings, we can use the formula:

```
=LEN(STRING)- LEN(SUBSTITUTE(STRING, LETTER,""))
```

To obtain the number of each letter in each line.

This works simply by counting the length, removing the letter, and recounting and the difference is the number of occurrences.

If we count each letter in each line we find in a matrix of each line we get the following:

Drop	e	a	s	l	o	n	g				Round results					f	->	r	Swap	
Line	b	c	d	h	i	j	k	m	p	q	r	t	u	v	w	x	y	z	output	end
9	0	0	0	2	0	0	1	0	1	0	4	5	1	0	2	0	0	0	T	:
8	1	2	2	0	1	0	0	0	0	0	3	4	0	0	1	0	0	0	S	.
7	1	5	1	1	2	0	1	0	0	0	1	4	0	0	1	0	0	0	C	.
6	1	3	3	1	5	0	3	0	1	0	10	2	1	0	2	0	0	0	R	.
5	0	0	2	1	1	0	1	0	1	0	5	0	0	0	2	0	0	0	R	,
4	2	2	2	1	1	0	1	0	0	0	1	4	0	0	0	0	0	0	T	.
3	1	4	4	1	5	0	1	0	2	0	4	7	0	1	0	0	2	0	T	.
2	3	7	1	5	5	0	3	0	2	0	5	7	0	0	1	2	1	0	C	,
1	1	3	1	7	2	0	1	0	2	0	2	4	2	1	0	0	0	0	H	.
Total	10	26	16	19	22	0	12	0	9	0	35	37	4	2	9	2	3	0	235	

We know that we are starting from the bottom and working our way up and have been told that in bitcoin at least, the winner is chosen by the one that produces the greatest number of hashes with most frequency will end up the winner. Then in the next round a new unique hash and thus block must be found, thus the previous winner cannot win this new round. This means that we must go line by line eliminating the last rounds winner as a contender.

The first thing we need to do before that is to create a matrix of sum totals of each letter, so that you can see what is going on clearly, we will stick to the format of the table above. Summing the count of each letter from line at index 1 generates:

Drop	e	a	s	l	o	n	g				Race Conditions					f	->	r	Swap	
Line	b	c	d	h	i	j	k	m	p	q	r	t	u	v	w	x	y	z	output	end
9	10	26	16	19	22	0	12	0	9	0	35	37	4	2	9	2	3	0	T	:
8	10	26	16	17	22	0	11	0	8	0	31	32	3	2	7	2	3	0	T	.
7	9	24	14	17	21	0	11	0	8	0	28	28	3	2	6	2	3	0	R	.
6	8	19	13	16	19	0	10	0	8	0	27	24	3	2	5	2	3	0	R	.
5	7	16	10	15	14	0	7	0	7	0	17	22	2	2	3	2	3	0	T	,
4	7	16	8	14	13	0	6	0	6	0	12	22	2	2	1	2	3	0	T	.
3	5	14	6	13	12	0	5	0	6	0	11	18	2	2	1	2	3	0	T	.
2	4	10	2	12	7	0	4	0	4	0	7	11	2	1	1	2	1	0	H	,
1	1	3	1	7	2	0	1	0	2	0	2	4	2	1	0	0	0	0	H	.

We could create a script that selects the winner and eliminates them from the next string but that is not very visual and overly complex for our purposes of explaining fundamentally how we have come to this output.

If we go through the entire table and eliminate the previous selected letter, we may well have more of an idea of what is being simulated and what else we might have left to do.

Drop	e	a	s	l	o	n	g		Tournament rules				f	->	r	Swap			
Line	b	c	d	h	i	j	k	m	p	q	r	t	u	v	w	x	y	z	output	end
9	-	-	-	-	-	0	-	0	9	0	-	-	4	2	9	2	3	0	P	:
8	10	-	-	-	-	0	-	0	8	0	-	-	3	2	7	2	3	0	B	.
7	9	-	-	-	-	0	11	0	8	0	-	-	3	2	6	2	3	0	K	.
6	8	-	13	-	-	0	10	0	8	0	-	-	3	2	5	2	3	0	D	.
5	7	-	10	-	-	0	7	0	7	0	17	-	2	2	3	2	3	0	R	,
4	7	-	8	-	13	0	6	0	6	0	12	-	2	2	1	2	3	0	I	.
3	5	14	6	-	12	0	5	0	6	0	11	-	2	2	1	2	3	0	C	.
2	4	10	2	-	7	0	4	0	4	0	7	11	2	1	1	2	1	0	T	,
1	1	3	1	7	2	0	1	0	2	0	2	4	2	1	0	0	0	0	H	.
Total	10	26	16	19	22	0	12	0	9	0	35	37	4	2	9	2	3	0		

While there is a tie in the last of the lines, we can't know if this line is even needed until we decode our message. Normally in this kind of list steganography the rules of the decode should be consistent they don't necessarily follow the same rules for each character. Due to the nature of a steganographic elements having to hide within other potential elements they each need to be uniquely positioned with unique instructions.

The method being used here is quite ingenious and works perfectly with its application, if we had no candidates for Satoshi, we would have a difficult time knowing if anything output was remotely plausible. A single consistent string of any letter wouldn't ever tell us anything. It could be done, but you would need to be very careful and took your time. Our result would be the same though as we will demonstrate characteristics like this are not easily generated even with large modifications to the intended method. Having a potential candidate helps but it cannot generate consistent steganographic rules in a text where there are none. This document as it turns out has many of these inconsistencies.

To us today, it looks like we aren't far off the intended solution. Especially as the rest of the puzzle has also indicated that he is the most likely candidate to have his name come out of a larger final element. Although there are a few letters [D, B, K] they have not in enough numbers to be included anywhere but the end of our puzzle after most letter have already been chosen.

We removed the A's as the instructions directed so it would seem that Adam Back, is no longer a possibility or at this point any other known contender for the Satoshi throne. If we are to perform some slight corrections due to packet loss, ordering, and receiving missed "transactions" it is unlikely that our current puzzle state will be able to generate anything other than what its author intended.

We have been told that we should expect messages to sometimes appear in the wrong order and have also seen that moving a comma after a period fixes an error, but which one to move may now be a little clearer. In this case line 2 ends in a comma and line 1 ends in a period, putting line 2 in the first position generates the first two characters [T]&[H].

Drop	e	a	s	l	o	n	g		Tournament rules				f	->	r	Swap			
Line	b	c	d	h	i	j	k	m	p	q	r	t	u	v	w	x	y	z	output	end
9	-	-	-	-	-	0	-	0	9	0	-	-	4	2	9	2	3	0	P	:
8	10	-	-	-	-	0	-	0	8	0	-	-	3	2	7	2	3	0	B	.
7	9	-	-	-	-	0	11	0	8	0	-	-	3	2	6	2	3	0	K	.
6	8	-	13	-	-	0	10	0	8	0	-	-	3	2	5	2	3	0	D	.
5	7	-	10	-	-	0	7	0	7	0	17	-	2	2	3	2	3	0	R	,
4	7	-	8	-	13	0	6	0	6	0	12	-	2	2	1	2	3	0	I	.
3	5	14	6	-	12	0	5	0	6	0	11	-	2	2	1	2	3	0	C	.
1	1	3	1	7	2	0	1	0	2	0	2	4	2	1	0	0	0	0	H	.
2	4	10	2	-	7	0	4	0	4	0	7	11	2	1	1	2	1	0	T	,
Total	10	26	16	19	22	0	12	0	9	0	35	37	4	2	9	2	3	0	2/9	

All the rest of the characters are roughly inline with what we would expect in terms of order with 4 of the early ones exactly matching what the puzzle has indicated to us.

```
<-- solving direction
PBKDRICHT
___WRIGHT
```

We can very quickly fix these problems with what we already have.

C is a letter that is the most like G, further G is one letter we removed due to the simplistic nature of the ruleset that we where following, if we take the more complex instructions that G can be left in the message, maybe it will be put in place.

Drop	e	a	s	l	o	n		Reinsert G						f	->	r	Swap			
Line	b	c	d	g	h	i	j	k	m	p	q	r	t	u	v	w	x	y	z	output	end
9	-	-	-	4	-	-	0	-	0	9	0	-	-	4	2	9	2	3	0	P	:
8	10	-	-	4	-	-	0	-	0	8	0	-	-	3	2	7	2	3	0	B	.
7	9	-	-	4	-	-	0	11	0	8	0	-	-	3	2	6	2	3	0	K	.
6	8	-	13	4	-	-	0	10	0	8	0	-	-	3	2	5	2	3	0	D	.
5	7	-	10	3	-	-	0	7	0	7	0	17	-	2	2	3	2	3	0	R	,
4	7	-	8	3	-	13	0	6	0	6	0	12	-	2	2	1	2	3	0	I	.
3	5	14	6	3	-	12	0	5	0	6	0	11	-	2	2	1	2	3	0	C	.
1	1	3	1	3	7	2	0	1	0	2	0	2	4	2	1	0	0	0	0	H	.
2	4	10	2	1	-	7	0	4	0	4	0	7	11	2	1	1	2	1	0	T	,
Total	10	26	16	4	19	22	0	12	0	9	0	35	37	4	2	9	2	3	0	4/9	

As we can see there are nowhere near enough Gs to be considered becoming a competitor without significant manipulation of this race. From removing a letter as per instructions to boosting it is likely one step too far.

```
PBKDRICHT <output
___WRIGHT <target
```

In place of the expected W, we have D. If we, you remember that the W is the only single starting letter of the line but also where we had the strange behaviour when copy and pasting much like

the hyphens. Of note that Craig Wright held a Doctor of Philosophy at the time making him Dr Wright. This could be an extra coincidence that sweetens the deal, or a red hearing to make us believe that we are done.

Many things indicate that the round should reset from this position, or something to that effect, the line before also has a comma which we have had to swap has indicated that instructions might be reset first lines around starts with a comma. This could indicate that it is a comma that denotes a block in our bitcoin simulation. Thus, we take the only non-repeated letter starting a line in the list, giving us:

PBK**WRICHT**
___**WRIGHT**

We are now left with a conundrum we can now clearly see the name WRIGHT, but we don't have any letters that make up his first initials in the top three contenders for the 2 positions we have from our checksum.

Now that we have exhausted the possibilities of the last name and have something we can confidently infer is a message communicated within the lines of the list we can reset the rules and see if we can obtain any remaining messages.

Letters that will not be shown in the coming tables because they haven't been contenders as they don't appear in any line in the list or very low down are [J, M, Q, Z, Y, X, V, U]. This is just so that we can fit the previously removed letters and restart the simulation. Considering that pervious work isn't carried over to the next round in bitcoin we need to restart the summation of each character from line 7 in our index of elements. After doing all this we can calculate the following table of frequencies:

Drop	g	j	m	q	z	y	x				Rules Reset									
Line	a	b	c	d	f	h	i	k	l	n	o	p	r	s	t	u	v	w	output	end
9	-	2	-	3	1	3	3	2	7		13	1	7	11	13	1	0	4	O	:
8	10	2	-	3	0	1	3	1	5	9	9	0	4	7	8	0	0	2	A	.
7	4	1	5	1	0	1	2	1	3	5	5	0	1	3	4	0	0	1	C	.
6		8	-	13	-	-	-	10	-	-	-	8	-	-	-	3	2	5	(W)	.
5		7	-	10	-	-	-	7	-	-	-	7	17	-	-	2	2	3	R	,
4		7	-	8	-	-	13	6	-	-	-	6	12	-	-	2	2	1	I	.
3		5	14	6	-	-	12	5	-	-	-	6	11	-	-	2	2	1	(G)	.
1		1	3	1	-	7	2	1	-	-	-	2	2	-	4	2	1	0	H	,
2		4	10	2	-	-	7	4	-	-	-	4	7	-	11	2	1	1	T	.

This first trial run has resulted in C once again becoming the first letter, we know this is supposed to be the letter S but this may simply mean we haven't got something quite right. We also have a three-letter tie which while we do have a potential method of organising seems excessive at this point in the game. The letter S is one we have filtered from the "As long as" phrasing, if we recall that the word before led us to believe that reintegrating the letter G would correct the C into a G. Unfortunately, or fortunately for us, depending on your perspective now, we can now estimate that the before long referred to the single letter before the word "long". That being the S in the word 'as'.

With this additional information we can infer that we should not reinsert all the letters, only the ones that we have confirmed by experimenting with what is and is not possible to reach our target via the instructions we are finding.

This update yields the following:

Drop	a	l	o	n	j	m	q	x	y	z	f	->	r	Swap	
Line	b	c	d	h	i	k	p	r	s	t	u	v	w	output	end
9	2	7	3	3	3	2	1	8	11	13	1	0	4	S	:
8	2	7	3	1	3	1	0	4	7	8	0	0	2	T	.
7	1	5	1	1	2	1	0	1	3	4	0	0	1	C	.
6	8	-	13	-	-	10	8	-	-	-	3	2	5	(W)	.
5	7	-	10	-	-	7	7	17	-	-	2	2	3	R	,
4	7	-	8	-	13	6	6	12	-	-	2	2	1	I	.
3	5	14	6	-	12	5	6	11	-	-	2	2	1	C	.
1	1	3	1	7	2	1	2	2	-	4	2	1	0	H	,
2	4	10	2	-	7	4	4	7	-	11	2	1	1	T	.

This is frustrating, although we now have the C and S by following the rules more closely, we are still missing something. With a T in the middle separating the two in the wrong order meaning this message still does not validate against the checksum found in the references. There are two interesting things about these letters that is worth considering. The C and the T have been used previously in our string. This maybe important since each letter is supposed to represent a block and blocks are uniquely identified we aren't correctly simulating the network.

The C has been utilised in place of a G which couldn't keep up with the others. This still meets the steganographic criteria but not the list, we are maybe able to give a bit of leeway on this letter if we find the correct solution.

The T has been used previously and although we needed to swap the lines around which was instructed by errors, allowed us to move on to the next part of the puzzle.

Generating possibly a final table last finally we achieve a complete string that is looking finished and save for a single manipulation while abiding by all the rules we have discovered in our steganalysis of the Bitcoin WhitePaper:

Drop	a	l	o	n	j	m	q	x	y	z	f	->	r	Swap	
Line	b	c	d	h	i	k	p	r	s	t	u	v	w	output	end
9	2	-	3	3	3	2	1	8	-	-	1	0	4	W	:
8	2	-	3	1	3	1	0	4	7	-	0	0	2	S	.
7	1	5	1	1	2	1	0	1	3	-	0	0	1	C	.
6	8	-	13	-	-	10	8	-	-	-	3	2	5	D	.
5	7	-	10	-	-	7	7	17	-	-	2	2	3	R	,
4	7	-	8	-	13	6	6	12	-	-	2	2	1	I	.
3	5	14	6	-	12	5	6	11	-	-	2	2	1	C	.
1	1	3	1	7	2	1	2	2	-	4	2	1	0	H	,
2	4	10	2	-	7	4	4	7	-	11	2	1	1	T	.

Remarkably we get even get ‘W’ as a final letter because we also removed the R like we did the T and as we were supposed to if we are being consistent. When writing a secret message in steganography honesty aids in the communication effort and in steganalysis, honest effort aids in feeling confident you got the right answer.

After some consideration it seems as though we could be more consistent in rule application as we have been. In this final step line 9 ends with a colon, and line 6 that we found began with D is immediately after a line that ends in a comma. As we have been interpreting commas as an error in our simulation that requires a swap operation. Then ‘:’ infers swap order of all elements the final string becomes:

Drop	a	l	o	n	j	m	q	x	y	z	f	->	r	Swap	
Line	b	c	d	h	i	k	p	r	s	t	u	v	w	output	end
6	8	-	13	-	-	10	8	-	-	-	3	2	5	D	.
7	1	5	1	1	2	1	0	1	3	-	0	0	1	C	.
8	2	-	3	1	3	1	0	4	7	-	0	0	2	S	.
9	2	-	3	3	3	2	1	8	-	-	1	0	4	W	:
5	7	-	10	-	-	7	7	17	-	-	2	2	3	R	,
4	7	-	8	-	13	6	6	12	-	-	2	2	1	I	.
3	5	14	6	-	12	5	6	11	-	-	2	2	1	C	.
1	1	3	1	7	2	1	2	2	-	4	2	1	0	H	,
2	4	10	2	-	7	4	4	7	-	11	2	1	1	T	.

Noting the final little feature D. as a title would be a DR. Resulting in the final message:

D. C. S. WRICHT <- Message

[C],[S],[W] <- CheckSum ✅

Alt-PoS.

It would be dishonest to say that: DCSWRICHT could not easily be interpreted to mean Dr C. S. Wright as a covert author of the Bitcoin: peer-to-peer electronic cash system. It would also be dishonest to say that we know for certain that there are not other messages within this paper that we have not found.

Having now completed the decode we set out to by finding a message, the as the person that found this messages, I am compelled to note that the first name I found was C.S. WRIGHT or some mutation thereof. On consideration of the method the author then attempted to create other names principally ABACK, ADAMBACK, Richard Hettinga or some permutation thereof. While it is in principle possible to come up with rules without looking for them. This is not a valid steganalysis method. Performing such an analysis would be considered in cryptography: “brute forcing”. A steganographic proof is not in the result of the final string like a magic artifact, it is in method and the process of extracting it. Like a metaphysical sword in the stone if can’t just be pulled out with force, if one does then they are left with explaining how they got the message, why they got the message and then must prove they can repeat the process. This needs a light touch to accomplish. The length of the message and its checksum validating the longer message is correct makes the chance of this being a feature of chance inconceivably low.

```
1234567890 123
DRCSWRIGHT.CSW
```

Just using the probability of the resulting we can estimate that brute forcing these 13 letters in order is roughly a 1 in 26^{13} chance of occurring:

That is $4e^{-19}$ or a 0.000,000,000,000, 000,000, 004%

Summarising our list of instructions:

- REMOVE
  - A, S, L, O, N, G
- SWAP
  - F -> R
- REPEAT
  - S
- SWAP LINES
  - 1,2
  - 7,8,9
- BLOCK FOUND
  - Line 6 indentation
  - Single starting letter
- SECONDARY ELEMENTS
  - Reference Key
  - Tape
- REFFERENCES
  - Names
  - Dates

While not an exhaustive list of instructions, missing characters, positions, spelling errors, formatting breaks in convention and consistency it helps illustrate just how many thing things had to come together to get to where we are.

If we are to look at the number of instructions and include them in our calculation, the number gets truly insane, requiring an additional 17 supporting elements means that the probabilities enter a region of **$3.5e^{-30}$ %** That number looks like:

0. 000,000,000,000,000,000,000,000,000,000,000,000,000,000,0035%

That process requires searching for errors and considering their possible interpretation as instructions to find the letters of a message. If someone attempts to find a message and then look for their instructions, they will likely find themselves making no progress. If someone starts with errors and attempts to try and miss interpret them, again, they will find themselves making little to no progress in the direction of the target that they intend to eventually end up with.

Using words or the formatting issue of hyphens to denote which letters should be filtered or amplified creates an interesting possibility not always present in steganography puzzles. It presents the opportunity of a dishonest actor to manipulation what would be considered reasonable interpretation of a result that the decoder wishes to find as opposed to objectively finds due to following the clues. The problem in doing this is that they would likely need to invent a rule and thus a corresponding mistake, an assumption about an existing rule that is not logical or ignore a mistake that we know is communicating a rule.

While adjusting the variable may yield some interesting results which will not be investigating them thoroughly. This is since even if we do attempt some brute forcing whatever we do will likely not be enough for some and the task will be never ending.

Since our decoding has highlighted CSW and appears to have crossed out Adam Backs name the author became curious that maybe some additional message was contained using his name. Searching of any possible permutations of Adam Backs name it was found that while many of the letters needed are contained in the lines in around the right area they are not in the required frequency or order to be meaningfully extracted via this method. Further, the chances of being able to pull two dissimilar names from the same paragraph using effectively the same methods is more than unlikely. The letter ‘m’ does not appear anywhere in the text and if we should expect “A Back” this leaves room for an additional few letters that may be interpreted as another instruction. Of some kind, yet nothing was ever found.

There are other names that fit a closer criterion for the list sets out, mainly the author of the first reference and a debated contribution by Adam Back, Wei Dai. They are also someone that has been speculated to be Satoshi Nakamoto, although they too deny these allegations. Our list contains all letters of their name and initials including the initial W the line of which under the rules we decoded resulted in D, the rest of the name being vowels limits this method of extraction with the clearly intended rules denoted by intentional errors. What is more both these

strings are smaller than our full result and as such by the rules of bitcoin are superseded by the longer string:

CSW DRICHT which when accompanied by two features resembling check sums in the network section list and the reference list resulting in CSW. The steganographic interpretation of the message is clearly intended to be DR CS WRIGHT.

I found that no matter what I tried I was never able to get a string of characters even remotely close to another name or initials that would be recognised as anything meaningful. I invite anyone to try and pull a different name out of this section, or any other section for that matter, utilizing a basic frequency analysis or steganalysis.

I did find that the last few lines of the Network section were able to quite easily generate the letters N, O, T which could infer that Wright is not Satoshi, however what is more likely is that this is a signal to the decoder that you can't reapply all the letters again, the rules must be applied consistently.

It is, I think, mathematically provable and logical that this is more robust proof than any cryptographic key and signed message could ever provide. A private public key pair can be transferred between owners. Steganography cannot be altered once it is created and published and thus cannot be transferred. That is why its most useful application is to covertly name things. And because it cannot be transferred, it is set in stone just like the protocol it describes.

-- During editing (09/03/2025) a friend and colleague in several ventures, John Pitts, who was extremely interested in my previous work in which I unmasked the founder and owner of the website AnonymousSpeech.org as Michael Gronager. As Michael Gronager he is the co-founder of the Crypto exchange Kraken, founder of Chainalysis who works with governments and crypto exchanges to trace criminal transactions on the blockchain. Otherwise known as Micheal Weber he was the founder of AnonymousSpeech.org a website to buy domains anonymously which Satoshi used to purchase bitcoin.org for launch. As Micheal Weber he is also the largest defendant but also a significant claimant in the Celsius bankruptcy case. Having withdrawn $55million of BTC and ETH within the 90 of collapse and then tried to seek $600,000 during the bankruptcy proceedings. These videos can be viewed on my X.com profile here.

John pointed out that an account going by u/Knockout_SS on reddit provided the information pointing to the companies bias in the Satoshi identity case Craig Wright was in the middle of. The company would have been able to verify the correctness of Craig Wrights claim and if verified would alter the makeup of the blockchain industry.

Due to the accounts posting activities pointing towards them being an avid follower of the developments of BSV, anonymousspeech and everything surrounding Craig Wright it might be safe to say that there is a non-zero

chance that this person is in fact Craig Wright. He does also use the same kind of speech anyone following Criag Wright is used to hearing “coreboy” being one of the core idioms he likes using. It is the account name here that is most interesting to us. Knockout_SS. In us decode we have found the filter that removes the letters needed to produce Craig Wrights name is between SS. A~~S LONG AS.~~

This is likely all the confirmation we should need that Craig Wright has been trying to subtly point people in the right direction and to the truth in the proof.

It is with hindsight that we can look back on what we have uncovered and reflect that it should have been obvious that the identity of Satoshi would be uncovered by simply looking for his mistakes. That is after all how a concealed identity is uncovered, but on this occasion, we should find it ironic that the mistakes that lead to the unveiling of the truth were not mistakes at all, but simply a misunderstood author.

In summary, it would be insane and illogical to believe that by running the rules of the Bitcoin network inferred from errors within the text that describes how it functions. That one would be able to retrieve any legible string by chance is beyond the realms of possibility. This feature must be by design. If we are to ask by ‘who?’ we need only read the output string and look at real world events and come to our own conclusions.

Conclusion. (Thoughts just after full decode)

The author of the white paper has demonstrated that they are a master in the art of communication no matter the technique and field. To create a paper with such impact and to so effectively conceal a message within it without anyone in public being able to tell for at least 16 years is not only a remarkable display of skill but also self-control.

Steganaography being a commination method is akin to a language that requires a logical truth in the steps it takes, be them visual similarities, geometric symmetry, choice in missing additional, or number of elements. These are all mathematical relationships that require an encoder and decoder to be honest in their effort to communicate and receive the same message. It is incredibly difficult to take a cover text and, without manipulation, create any string of characters from analysis out of what might be considered an error. It is even more difficult to create a message that has a meaning within the context of the medium it is communicated within.

If you have read this analysis in full you should now understand the concept of why it is a fundamentally different but equally rigorous process used often to conceal messages. If you want to compare steganography with today's technology each message is like a little computer program written in its own language where an out of place comma is the difference between a cover and the message the messenger genuinely wished to send. If you don't understand why this is an effective and secure method, then I ask you contemplate the cover of this report and contemplate why it is the way it is. To someone that has yet to read the report and the seemingly random order of punctuation it should be plausible that I have hidden a message within it that only those that have read the report would be expected to correctly interpret. I would expect that by now, and maybe after a second read that with a quick glance and a second look, you might be able to piece together what is being communicated to those squiggly lines on the page.

While we can't be certain that other solutions don't also exist, as is the nature of this scientific process of experimentation utilizing existing frameworks and previous articles of work can't explore every potential combination of rules. We may be able to turn once again to the paper for some insight as to how to deal with this conundrum. Within, it explains that we should take the first seen and longest chain to base our assumption of honesty on and thus integrity of everything is hosts. So far it is the only steganographic solution found and so it is the longest and first seen. I have tried my best to make other names appear but being versed in steganography I know that it is hopeless. If you don't think it is hopeless then you should, try and make this list or another area of the paper say another name in steganography that is longer and more fitting.

To the author of this report and any honest person reading it should be irrefutable proof that Dr C. S. Wright (C.S.W) authored the Bitcoin White Paper. Furthermore, Rev. Dr. Wright went to great lengths to make a point about Adam Back.

From being the first person known to have received an email from Satoshi to check the reference that was used to then, cross out that reference possibly after Adam chose not to attribute Dwork and Naors paper. Yet there is something more sinister about this situation, the

service Satoshi used to deliver the PDF to Adam Back was covertly advertised on the CypherPunks mailing list in 2005 using steganography. Craig Wright using that service was clearly aware that this 'CP' group utilised and had good knowledge of Steganography. This could of, and likely has, opened him up to attack by the group without anyone being publicly aware.

The author being suspicious of their activities has worked hard and diligently to point to Adam Back as a dishonest individual. There is clearly an ongoing dispute over this technology that need to be addressed publicly. The U.S. Government, which at the time of writing, is currently organising their economy around technology based on this White Paper and thus BTC is acting under false pretences.

On one side you have Craig Wright who was prematurely out-ed as Satoshi Nakamoto by W.I.R.E.D and Gizmodo in 2015 which they later retracted after pressure. The fall out of this event resulted in Bitcoins single global chain splitting, not once, but TWICE (2017, 2018) with each time-chain (timeline) claiming to record events according to the Bitcoin rules set forth in this paper.

Contrary to popular belief Craig Wright did not participate in the mining of BCH during the time of the first split in 2017, a mysteriously hosted node in Asia with the hash power of a miner initiated the fork on a technical level.

On an economic level BitFinex provided customers trading liquidity to bet on the announcement and its code published by Amaury Sechet. No matter how hard they may try not to be, these lead advocates end up being the de facto leaders of each resulting chain, those being:

		Bitcoin Ticker		
0.	Satoshi Nakamoto			
1.	Craig Wright, et al.	BSV	<-	Restore original protocol
2.	Roger Ver, et al.	BCH	<-	Add functionality
3.	Adam Back, et al.	BTC	<-	Add functionality

There is a common saying in the military, once is an accident, twice is happenstance. Three times is enemy action. With three competing chains and claims to the title of Bitcoin which this paper outlines, the path of Craig Wright most closely follows the genuine narrative set out by the errors within the paper. Either this is the voice of GOD for some reason pointing to Craig Wright as a Manchurian Candidate or, more logically, Craig Wright is, in fact, who he claims to be. Satoshi Nakamoto.

For all the hype there are several things that Bitcoin can and cannot do. It is not an oracle. An oracle outputs a truth from an input in a defined single step. Bitcoin cannot do this, truth is not a single even, it is many stacked on top of one another to produce an outcome. Thus, no machine can ever be an oracle because a machine no matter how hard you may try to program it to determine what is true and what is not. A machine is bounded by their own limits and cannot perform this task, humans struggle with it how could we ever expect a machine to understand truth from fiction. Bitcoin practically demonstrates any this. It will record whatever you pay for. It's the source that is the oracle, and the source isn't a machine, it is a being.

A machine is a physically limited mechanical systems that come to a result, not determines the truth, only humans, without have the ability to reason through a creative non-random process

can accomplish such a thing. We can only interpret what is recorded for ourselves, turning seemingly random data into information for others to consume as clearly as we attempt to communicate it.

As I contemplate how the world might change after reading this information, I can't help but wonder, what would have happened if I had not have found it. Are people even waiting or expecting it? I have a suspicion that the people this technology threatens, and targets know all too well that this steganography proof exists, I have reason to believe that they have already decoded it and know that Craig Wright was Satoshi Nakamoto. In fact, I have reason to believe it is the entire reason he currently has been slapped with a suspended prison sentence forbidding him to claim to be Satoshi Nakamoto under threat of further imprisonment and fines. This injunction currently has prevented him from commenting on anything related to this decode during writing this document. At the time of publication, he is unable to discuss why his name is embedded within the paper using steganography. He is the only person in the world to have his speech restricted in this way and to everyone but him, this is supposed to be new information. This circumstance should give everyone with the knowledge in this report a reason to double check their core assumptions of the blockchain and cryptocurrency industries.

To end my conclusion, I would like to explain something about communication using a quote from an unknown Zen master, this quote while used by Alan Watts many times has a much more subtle meaning to the military and intelligence agencies, they hope to instil their troops with an idea, and that idea is action with as little communication as needed or as a Zen master puts it: **The sound of rain needs no translation.**

What happens next?

Phase 2 of bitcoins life, hopefully, it should be leaving its foolish years of adolescence and becoming a stable foundation that the world can build its systems within.

Many of us are probably wondering what the man himself would like to say about this. Well as per court instructions he has not answered a single question I have had with regards to the origin of bitcoin since the ruling against him was passed down. I feel that this is a great loss of talent and is an injustice to all those who are currently falling victim blockchain and crypto shams and scams. The people that are enriching themselves are currently surrounding each countries governments hoping to provide them with their own flavour of the world changing invention that doesn't scale. Except, its creator has now scaled his vision to be more capable than any other core banking system in existence. For all the talk of centralisation vs decentralization, AS LONG AS the protocol doesn't change it won't be LONG BEFORE everyone has gotten used to the way the new things are.

Dr Craig Wright through his account on X.com @CSTOMINAGA is at least aware that this decode has happened by an independent third party. Although he hasn't commented on it directly, he has made a few posts which I see of words of encouragement to me while I attempt to explain his actions:

Mar 7, 2025

```
Writing is a slow reckoning, a confrontation between what is and what could be. It begins
as an impulse, a fragment of something unfinished, clawing its way to the surface. To write
is to carve into silence, to demand meaning from absence. It does not come easy. It should
not. There is nothing easy about peeling back the skin of a moment and pressing it onto a
page. A story that lingers is never just a story—it is a reckoning, a weight, a thing that
refuses to be ignored.

Style is not something that can be chosen. It is revealed, slowly, through the act of
writing itself. What has changed is not my style, but my understanding of it. The silence
between sentences has grown heavier. Dialogue is not just words exchanged—it is a
battlefield, a negotiation. There are no wasted lines. Every breath matters. The air
between words must carry its own weight, thick with what is left unsaid. In this, I have
learned restraint. The tendency to explain, to guide the reader too closely, has been
stripped away. What remains is sharper, leaner, unwilling to give more than is necessary.
The best writing does not lead; it pulls. It demands that the reader step forward into the
dark, unbidden.

Readers are unpredictable things. They bring their own ghosts, their own histories, and so
they do not always see what was intended. This is the risk of writing, the inevitability of
it. Some saw the war as backdrop, the romance as peripheral, but this was never just a war
story, never just a romance. It was about consequence. About the weight of a single
decision. About the way silence can speak louder than a bullet. Some understood. Some did
not. That is the nature of stories. If I have learned anything, it is that the writer does
not control the reader's interpretation, only the precision of the world they create. The
rest belongs to the space between the lines.

Writing should not be easy. It should bruise. It should ache. That is what makes it worth
reading.
```

Improvement comes in increments, in fractures of understanding that widen with each rewrite. The pacing is better. The weight of a moment is clearer. The silences linger longer. But the flaws are still there. They always will be. The internal voice of a character—what is held back, what is revealed—this is where the work remains. The psychology of a moment is more delicate than the action itself. A gunfight is not interesting. A man's hesitation before he pulls the trigger is. That is where the next stretch of writing will be focused—deepening the internal war, ensuring that when a decision is made, it is felt, not just read.

Textbooks do not matter. It is the voices of others that does. The discussions, the arguments, the critiques—this is where writing grew. Writing is not theory; it is practice. It is sitting in the dark with a scene that does not work and pressing into it until it breaks open. No book can teach that. Only the act of writing can. If there is any advice to be given, it is this: writing is not about knowing. It is about discovering. Do not seek to control the story. Let it unravel. Follow it. And when you think you have reached the end, go deeper. There is always something beneath.

The story is not finished. It never is. Even when the last word is written, the characters keep moving, their choices reverberating beyond the page. A good story does not close a door. It leaves it slightly ajar, the draft whispering of what still lingers beyond it. That is what I hope this screenplay has done. Not to answer, but to ask. Not to settle, but to unsettle. Because writing is not about what is given. It is about what remains.

possibly the greatest most public and most real magic the world has ever seen. It wasn't a trick, it was merely happenstance and strategy, the stakes are most definitely real.

Now that humanity has received and hopefully accepted this new way of looking at old information everyone is probably asking, where do we go from here. When you realise that no one can tell you that and that is all up to us individually to make a collective decision on which path to take with the future of humanity, and that choice will be recorded in the immutable nature of us. Then I trust that everyone will come to the same conclusion without the need to communicate to strongly about what they need to do.

The concept of an anti-meme is becoming more and more meme like. The theory many people are working with is that for every anti-meme that come into existence there must be a set of fake memes that proliferate society as a silent wink to those in the know. When an anti-meme is exposed, its fake meme is destroyed along with its anti-meme. To create a fake meme, it requires an anti-meme in which those that create the fake memes know they are lying and wish to slip an incorrect fact into the cultural zeitgeist thus fooling others into participating in the spreading of the fake news and meme. If they come into close enough proximity, they cancel each other out. If the truth is ever exposed, a real true meme takes the place of both. A lie creating a paradox within what is perceived to be true, when it is compare directly and in an unbiased way leads to the annihilation of the imposter. At least in most of the minds of readers.

When I announced that I had performed the decode of the paper to the current small group

Reality is what you can read, but is that reality?

I'm not sure that words can really describe the critical nature of the situation. The world is corrupt. So, corrupt that something as pure as bitcoin, in its meticulous design and its clearly communicated goal, or its carefully considered launch; was ripped apart piece by piece in public and no-one felt like they could do anything about it. The only thing that prevented it from going further down the darkest timeline was the unveiling of its creator.

I’m also not sure what the future holds for me, I guess it depends on how this information is received. At the moment, although this is a positive development for me and my way of looking at the world unless this information changes something I’m

Additional thoughts in editing room.

I expect I am coming to the end of my own personal editing and addition of paragraphs while proof reading is taking place by my most valued supporters.

When trying to decide how to sign this document I, of course, couldn't help but think and possibly feel what Satoshi might have gone through although at his time the stakes were much greater. Time having moved forward and changed the world quite considerably since 2008, I am already quite publicly known in the Bitcoin industry, at least to my fellow enthusiasts. So, when I decided to start to turn these notes into this report, I had to make a choice; "do I tell others? (or not)". I didn't think twice. ", or not" never had any staying power in my mind or heart. It was only a question of how to communicate it effectively such that this matter be put to rest, unequivocally. As such I likely don't have the same liberty as they Dr Criag Wright did in publishing this under a name that is different from my true name. The name I was born with, and like for most people, is ingrained so deeply in every fibre of my being in such a unique way that I don't think I could ever be separated from it. Given that this topic has been shrouded in mystery for so long and it's my reputation amongst these enthusiasts that means they are willing to support and assist me with this report. It would be unfair to their contribution to topics dependent on these early events that they have all helped uncover and distribute so publicly with their own reputation at stake. A risk they took, not just over the past few hours, days or even months, but years. One and a half decades. They along with myself, need some closure on this topic at least, once and for all.

Everyone has always assumed that Satoshi was using a pseudonym wishing to hide themselves from those that control the money supply/big tech or anyone else that bitcoin threatens. Effectively because they feared the consequences for themselves. On this journey I had a completely different perspective that I'm almost certain is the case because it feels so universal to us all, heightened due to the accelerating technological progress that can feel oppressive. I believe that Dr Wright did things the way he did not out of fear for himself or others around him, but because he needed to create a cry for help that everyone could relate to without it being muddied by the reputation of the person that blew the whistle. In this way, he didn't do it for privacy reasons, likely knowing that no matter how hard he might try he would most likely be found out anyway, he did it so that we might be able to help ourselves however long it might take for us to have the motivation to go out and look in the right places when we know enough about how things work.

It is not HIS CRY FOR HELP we hear echoing in the darkness,

it is ALWAYS OUR OWN.

We need to be comfortable owning that as a group that calls themselves at least in this language Hu-man.

About the Author

Supporters and contributors' names as they wish to appear and the order they did:
[assumed based on usage in BSV get in touch if not correct]

Editors: Morht, Joel Dalais, Todd Price, 1 (@369BSV), Darren Kellenschwiler, Jorge Pelaez, Jerry David Chan, Ken Shishido, KuzzyBro, John Doe, Thomas Jacobson, Timothy Swan, Fredderic Honohan, Duane, Erik van Velzen, Brenden Lee,

Copyright notice

Legal

Conflicts of interest statement

Further work

Sections 3,6,8,9 ,11 and 12 don't seem to have any steganographic manipulation to speak of all though the longer I look the more I'm seeing, especially with respect to spell check. There maybe things that I have missed and are worth going over with fresh eyes. The list might say something else. Even if small to guide someone to the correct result. To do so would first require creating the words that they want to embed and then check that they have similar letters in similar positions. Then try and construct a third set of sentences that is legible and has both sets of letters pop out for independent rules. This sounds like to tall a task. But knowing Craig Wright, I know he wouldn't bother as it sounds an awful lot like a three-body-problem.

Until someone has found the rules who is to know?

We have found evidence that Dr Craig Wright may have used WhiteSpace Steganography software to mark the paper as his own or send a different secret message. This is worth investigating and if done, recommend that the investigator start simple and think fundamental, know the field material and check our source material.

Might be nothing.

Signed:

■ Fauvel (?)

L.

Appendix A.

Overview of most basic method.

Main message:
Element: List in Network Chapter 5
Reason: Final number indicated by reference element.
Uses: Decode for full message.

Inferred instructions:
Punctuation infers items such as blocks. -> Find a block, find letter of message
Missing hyphen in non-reversible -> Start at end of list
Comma misplacement – (received), (found) -> Replace f with r
“The tie will be broken” }
“becomes longer;” } -> Swap lines denoted by ‘;’ and ‘:’
“switch to the longer one” }
Repeated words “As long as … before long.” -> Remove ALONG, Recover S in round 2
Comma is a round start -> New round remove summation filters
“Packet loss” }
“Dropped messages” } -> at least 1 letter is not correct
“Realises it missed one”. }

Applied Result and method:
Cumulative Frequency analysis of letters between each punctuation point starting from end with applied filters and resetting at line 3.

3.D) Each node works on finding a difficult proof-of-work for its block.
2.C) Each node collects new transactions into a block.
1.S) New transactions are broadcast to all nodes.
0.W) The steps to run the network are as follows:
4.R) When a node finds a proof-of-work,
5.I) it broadcasts the block to all nodes.
6.C) Nodes accept the block only if all transactions in it are valid and not already spent.
7.H) using the hash of the accepted block as the previous hash.
8.T) Nodes express their acceptance of the block by working on creating the next block in the chain,

Interpretation:
WSCDRICHT => D.CSW WRIGHT
Transmission errors: G missing its internal line so appears as C.

Message CheckSum:
Element: [7][5][2]
Reason: out of order against convention
Uses: Chapters of interest and letters of checksum within references

Applied Result:
References: find characters in position of indicated by next reference (ROT1)
[7]: C (position 2)
[2]: S (position 5)
[5]: W (position 7)

Chapters:
[Key] [7]: where the element was found
[-] [2]: Missing hyphen in otherwise hyphenated word “double spending”
[List] [5]: list of steps to run the network

Puzzle completion check:
[10 (binary for 2)]: time and size of individual trades, the "tape"
Leads to references [7],[5],[2] back to checksum/key.

Appendix „B”

Text of the BITCOIN WP, all formatting text, punctuation maintained except for line breaks and equations such that it is more legible.

Author of this report has highlighted errors used as instructions and results where appropriate.

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

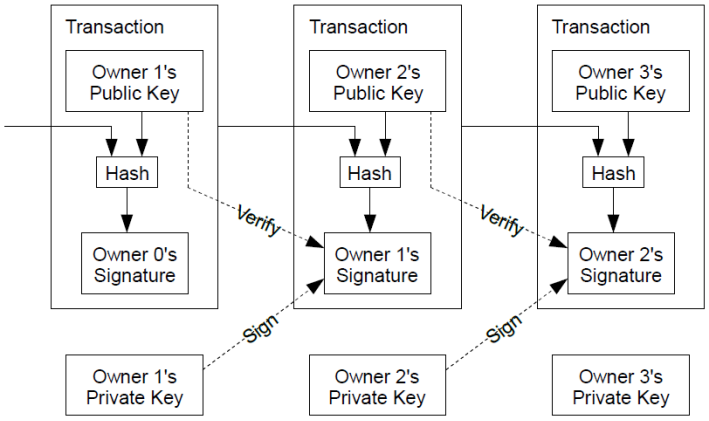
Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

page number -> 1
--- page break ---

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner ransfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

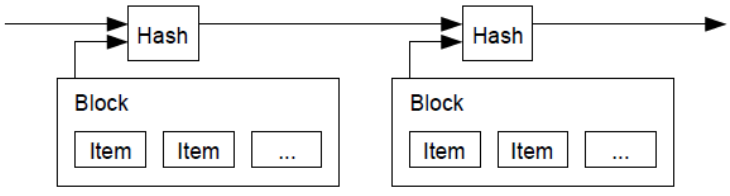


The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

3. Timestamp Server

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

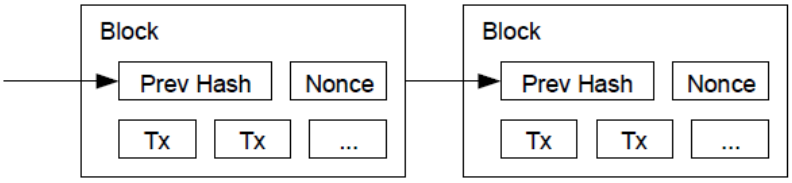


page number -> 2
--- page break ---

4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proofof-work system similar to ~~Adam Back's Hashcash [6]~~, rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

C) New transactions are broadcast to all nodes.
S) Each node collects new transactions into a block.
W) Each node works on finding a difficult proof-of-work for its block.
RI) When a node finds a proof-of-work, it broadcasts the block to all nodes.
G) Nodes accept the block only if all transactions in it are valid and not already spent.
HT) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proofof-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

page number -> 3
--- page break ---

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

6. Incentive

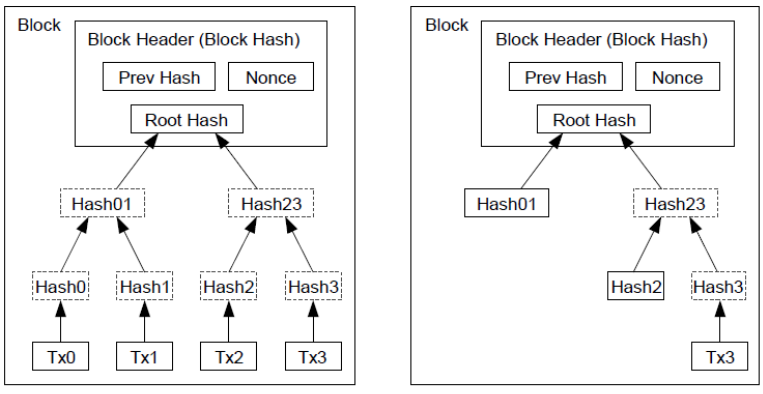
By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is **CPU** time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more **CPU** power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

7. Reclaiming Disk Space

Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.



Transactions Hashed in a Merkle Tree

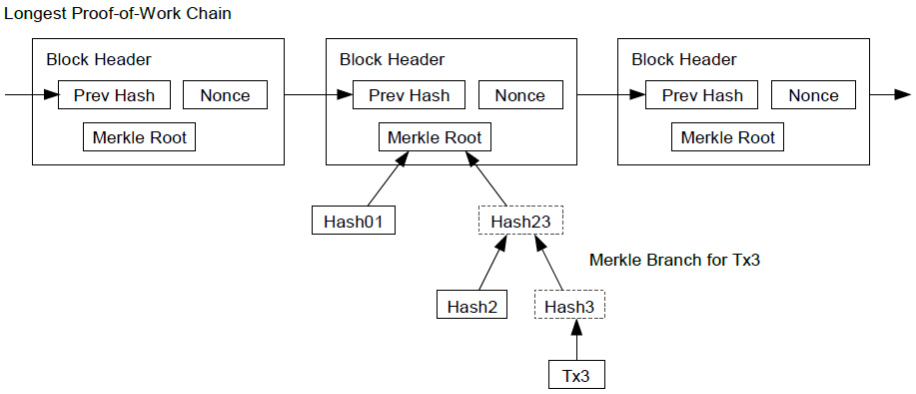
After Pruning Tx0-2 from the Block

A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

page number -> 4
--- page break ---

8. Simplified Payment Verification

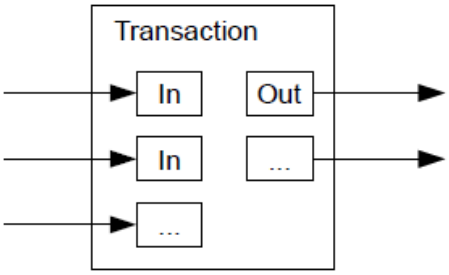
It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. Combining and Splitting Value

Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.

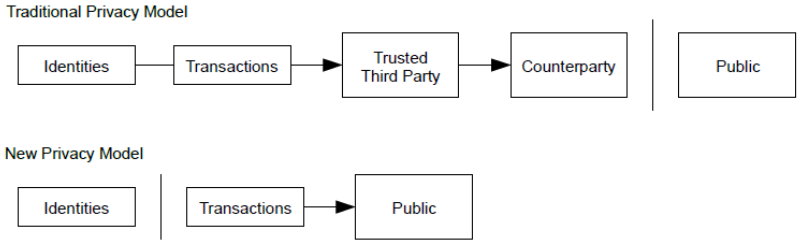


It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

page number -> 5
--- page break ---

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

11. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind
q_z= {1 *if* $p \leq q$, $(q / p)^z$ *if* $p < q$} *<- formatted for reading*

page number -> 6
--- page break ---

Given our assumption that $p > q$, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$\lambda = z(q/p)$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \lambda^k e^{-\lambda} / k! \cdot \{(q/p)^{z-k} \text{ if } k \le z, \ 1 \text{ if } k>z\}$$

Rearranging to avoid summing the infinite tail of the distribution...

$$1 - \sum_{k=0}^{z} \lambda^k e^{-\lambda} / k! \cdot \{1-(q/p)^{z-k}\}$$

Converting to C code...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
	double p = 1.0 - q;
	double lambda = z * (q / p);
	double sum = 1.0;
	int i, k;
	for (k = 0; k <= z; k++)
	{
		double poisson = exp(-lambda);
		for (i = 1; i <= k; i++)
			poisson *= lambda / i;
		sum -= poisson * (1 - pow(q / p, z - k));
	}
	return sum;
}
```

page number -> 7
--- page break ---

Running some results, we can see the probability drop off exponentially with z.

```
        q=0.1
        z=0 P=1.0000000
        z=1 P=0.2045873
        z=2 P=0.0509779
        z=3 P=0.0131722
        z=4 P=0.0034552
        z=5 P=0.0009137
        z=6 P=0.0002428
        z=7 P=0.0000647
        z=8 P=0.0000173
        z=9 P=0.0000046
        z=10 P=0.0000012

        q=0.3
        z=0 P=1.0000000
        z=5 P=0.1773523
        z=10 P=0.0416605
        z=15 P=0.0101008
        z=20 P=0.0024804
        z=25 P=0.0006132
        z=30 P=0.0001522
        z=35 P=0.0000379
        z=40 P=0.0000095
        z=45 P=0.0000024
        z=50 P=0.0000006

Solving for P less than 0.1%...

        P < 0.001
        q=0.10 z=5
        q=0.15 z=8
        q=0.20 z=11
        q=0.25 z=15
        q=0.30 z=24
        q=0.35 z=41
        q=0.40 z=89
        q=0.45 z=340
```

12. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

page number -> 8
--- page break ---

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.
[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.
[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.
[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.
[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.
[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.
[8] W. Feller, "An introduction to probability theory and its applications," 1957.

page number -> 9
--- End of Document ---