BIRMINGHAM CITY
University

CMP6200/DIG6200

INDIVIDUAL UNDERGRADUATE PROJECT

2023–2024

A2: Literature Review and Methods

AUDICRPYT: A CRYPTOGRAPHICALLY VERIFIABLE LEDGING SOFTWARE FOR LOCALISED, SENSITIVE ENVIRONMENTS – WITH BLOCKCHAIN TECHNOLOGY

Course: MSci Integrated Master of Computer Science
Student Name: Matthew Potts
Student Number: 21124021
Supervisor Name: Shadi Basurra

Contents

1 Report Introduction

At a brief perspective, this review documents my literature bibliography into academic sources related to my project, furthering context into justifications I've made for implementing specific features into my program. I intend to detail my research methodology, and criteria I operated with to maximise information gain from sources; likewise, to understand – at a wider perspective – how each source relates to my project – I've enlisted a theming system to categorise each source. By implementing a system like this, one can comprehensibly understand how and why I'll implement specific aspects of my software in the coming months.

Furthermore, I'll go on to document a fundamental design plan my software will align to, coupled with a discussion into a design methodology I intend on operating with – in this case, a Spiral-like modus operandi. Finally, I'll explain the testing regime I intend on implementing towards the rear of this report; in conjunction with a user requirement criteria earlier on.

1.1 Aim and Objectives

A comprehensive list of project-spanning objectives has been detailed below:

- Create cryptographically-verifiable receipts for every data transaction in a Local Area Network; with a mixture of SHA-256 hashing, and AES-256 encryption techniques. Ensure this accounts for both local institution data policy, and national data legislation.

- Create a data recording mechanism capable of detecting, and analysing, specific file changes; dictate whether they're either for malicious or clean purposes. Ensure ledger is cryptographically intact, immutable, and consistently in-line with relevant directories.

- Establish secure, encrypted LAN connectivity utilising a mixture of symmetric and asymmetric cryptography.

- Provide each new client in a decentralised network with entirely separate digital signatures, can be traced, and verified for authenticity.

- Ensure UAC hierarchy is installed, ensuring team of moderation (albeit within restrictions due to possibility of 51% attack); is present.

- Provide enterprise-grade data policy integration, in-line with both national and international data legislation relevant to enterprise location.

- Integrate logical, and temporal filters, into file transaction history – to complement malicious activity investigations, and identify patterns with specific users in question.

- Although not an essential at this moment in time, but I do envisage implementing a graphical interface into this software at some point.

1.2 Literature Search Methodology

Initially, having begun an initial literature search into this topic – I came across a few keywords encapsulating the general premise of this project; including terms like "Cryptographic Integrity", and more relevantly – "Cryptographic Integrity in a Data Ledging system". Although this provided me with a decent entry point, it somewhat distinguished itself into similar themes – including "Blockchain", and its parent technology: "Distributed Ledger Technology" (DLT). Whilst researching into the anatomy of concepts such as Blockchain's and DLT's, I researched into "Tokens", similarities in cryptocurrencies (i.e. their blockchains), and how they can be cryptographically tied, in a linear fashion, similar to that of a "Ledger". A key aspect of a blockchain is that individual blocks comprise of 2 corresponding components; this being a "nonce", and a "hash". I feel I could implement a similar anatomy into receipts of mine to ensure they're cryptographically verifiable. I've also researched into "Cryptographic Signatures", these – often implemented in a DLT – formulate a basis, as briefly mentioned earlier, for asymmetric cryptography for both sender's and recipients in a network. Throughout this, I strictly maintained my research methodology to utilise only high-end academic repositories – where I could leverage the authenticity, and relevance, of a source – with many different parameters; including how many people cite it, and what type of source it is for instance; newspaper articles, to name an example, aren't as reliable as an academic paper by a respected institution. A couple of academic repositories I used include BCU's own Library Search (See Appendix One), and IEEE Xplore (See Appendix Two).

I've ensured I also had a dig into similar real-world case studies. A few examples of ledging software include Amazon's "QLDB" service; and Google's "Trillian" service. Alternatively, a prominent theme I had a dig into also comprised of Malware into DLT's, Blockchain's, Ledger's, and to Cryptography in a nutshell. Similarly, I'd also delved into malware associated with both LAN formations, intertwined with encrypted blockchains in sensitive environments; producing findings on risks like a "51% attack", and "double spending", this can be tied in with "redundancy" – a common issue in both Ledging practices and databases.

A key point of this software is that enterprise "data policy"; and national "legislation" is maintained. A crucial aspect of this is that it can be tailored specifically to individual enterprise requirements. Throughout this literature research process, the topic of an AI integration has crossed my mind – specifically, "analysing data trends", and "user blockchain behaviour" – to predict whether file changes are of malicious intent. This, being somewhat advanced – isn't of an integral nature to this project; though it can vastly improve data security by analysing users early on for suspicious behaviour, and using developed "classification" techniques, and models, to evaluate their intentions on specific files, and workflows.

All aspects of this software align with specific "protocols", and in tandem with similar "validation" techniques. Although this section covers an adequate majority of terms used in my search – instances of others do exist in brief contextual searches – these haven't been mentioned for sake of being concise.

2 Literature Review

2.1 Themes

Most importantly throughout my literature search, I've employed a thematic approach to distinguish sources into areas of integral topics; e`ssential to fulfilling a full and comprehensive literature review of this field. I've used 8 themes to accomplish this, for instance – a theme includes "Ledging" – a practice, as described earlier, implemented in numerous environments, like banking, and data collection; a few sources aligned include, for example, the likes of Amazon's" "QLDB" (Amazon Web Services Inc, 2021) and Google's "Trillian" (transparency,dev, 2021). A collection of Cloud-based services; they're monitored, and use quotas which limit performance vastly – in contrast to my project, being open source, it'll be entirely customisable with no quotas being enforced on nodes that can join a LAN, and other features for instance.

Similarly, another theme is "Blockchain", which I've taken significant inspiration from. Although a little broader than those just mentioned, this is due to me looking into real-world case studies regarding Blockchain – including malicious events, and how specific configurations, in complement to the framework of a Blockchain itself – has promoted successful software developments in past years; an example of this being "Blockchain in Card Payment Systems" (Godfrey-Welch, Darlene; Lagrois, Remy; Law, Jared; Anderwald, Russell Scott; and Engels, Daniel W, "Blockchain in Payment Card Systems, 2018) – whereby blockchain aid in providing a "single tamper-resistant digital ledger", coupled with "hash pointers used to record encrypted transactions in a structured manner". My research into Blockchain framework's also rooted further particular research into similar framework's such as Hyperledger Fabric, as described in the last section (wiki.hyperledger.org, n.d).

Likewise, two similar topics I've researched into include both Malware, and Cryptography. Regarding the latter, more specifically implementing "Access Control into a Distributed File System" (Harrington, A. and Jensen, C, 2003, June); where concepts including "client-side'" encryption, combined with both asymmetric and symmetric cryptography; and synchronised server, and client-side authentication –

where data is only transmitted once provided encryption methods are completed on both machines. Similarly, Malware intertwines well into this; providing concrete evidence that no Blockchain, regardless of precautionary measures – is entirely secure. Instances of “51% attacks” – where “a group of miners control more than 50% of the network’s mining hashrate, or computing power” (Ye, Congcong et al. 2018) – have enormous repercussions if exploited. Due to my project’s inherently sensitive nature, it’s of the highest priority to ensure vulnerabilities, as such, are comprehensively ironed out before release.

Finally, Artificial Intelligence constitutes a final theme of research; more specifically regarding how forms of AI could be integrated within my project to develop greater comprehensive data analysis – dictating whether specific user activity can be regarded as malicious; as touched on before. For context, I investigated Villaim Langsch’s “Speechless: A Virtual Personal Assistant” – where I used concepts like their “Speech Recognition Engine”, in its object-orientated framework, as an ideal cornerstone for AI usage in my project (Langsch, V. “Speechless a Virtual Personal Assistant.”, 2018).

2.2 Review of Literature

2.2.1 Review & Theory

For sake of organisation, I’m going to divide this section of my report into further subsections – each explaining references relevant to particular themes I acknowledged briefly just before. To keep matters further concise, I’ve integrated any relevant theory into respective subsections.

Ledging:

Firstly, I investigated similar case studies, in relation to ledging – with integrated cryptographic technology involved in some fashion – either in SaaS distribution (i.e. Cloud), or as standalone applications; furthering into both AWS’s “QLDB” or Google’s “Trilian”. Progressing in regards to “QLDB, or “Quantum Ledger Database”; otherwise described as a “fully managed ledger database”, providing “transparent, immutable, and cryptographically verifiable transaction logs” (Amazon Web Services, Inc, n.d.) - being a Software as a Service (SaaS), it’s capable of maintaining a “sequenced history of every application data change using an immutable and transparent journal” – with “ACID”-enforced integrity. “QLDB” serves significantly in providing indispensable inspiration for my project, including a benchmark for what features should be available, how it could develop, and how mine can distinguish itself as a practical alternative for a wide demographic looking for greater data security options.

In terms of how it operates, it shares similar taste in hashing, and cryptography methods; employing SHA-256 hashing with QLDB’s bespoke API to prove integrity of any data change. To maintain zero data redundancy, given that this is in database format – “QLDB” enforces Full “ACID” checks; ensuring “transactions have full-serializability”, “atomicity…,isolation, and durability properties” (Amazon Web Services, Inc, n.d.). Albeit, it claims QLDB can “easily scale up and execute 2-3x as many transactions as common blockchain frameworks”; I have reason to dispute – as numerous sources claim that despite evidently slower read/write speeds of a blockchain framework; “blockchains are faster than databases when it comes to verifying transactions” (sanvignesh, 2023), although this is discussed in greater depth in the next section.

An issue I had with “QLDB” was its quota requirement, albeit expected given it being an SaaS. By default, “QLDB” only supports 5 “active ledgers, and only 5 “active journal streams to Kinesis Data Streams” (docs.aws.amazon.com, n.d.). Although useful enough for most use-cases - particularly large environments, or those with a high volume of sensitive filesystems, to name a few examples, will require greater scope for expansion then AWS provides, and this can become an exponentially expensive investment – and one that not all will be prepared to make.

Similarly, I investigated Google’s “Trillian” software; being “open-source” – its entire source code repository can be visited on GitHub (GitHub, 2023) – or viewed in Appendix Four. “Trillian” can be adapted to countless environments; with “tamper-evident logs”, “Trillian” can provide an entirely “verifiable system” (transparency, n.d.). As I mentioned before, in describing my project’s key objective in maintaining enterprise-level policy, and national and/or international legislation, compatibility – this took inspiration from “Trillian”, describing itself as a tool to “simplify regulatory compliance” (transparency, n.d.).

Alternatively, Trillian’s GitHub repository indicates that it differs from similar cryptographically-verifiable ledging software, it also details it uses a “data storage layer, to allow scalability to extremely large trees”, more specifically – “Merkle trees” (GitHub. 2023). For context, “Merkle Trees” are “used in distributed systems for efficient data verification”, by using “hashes instead of full files”, they’re far more efficient (Brilliant.org, 2016); in combination with modern SHA-256 hashing, they’re much more secure. In organisation terms, given that Merkle Trees share common features with Binary Trees, it ensures computational simplicity in rapidly traversing for particularly hashes.

Blockchain and Distributed Ledger Technologies (DLT):

By far the most significant aspect of all those included here; my research into both Blockchain’s and Distributed Ledger Technologies (DLT) comprise a relatively large proportion of my literature research. First and foremost, I wanted to analyse specific anatomic features within Blockchain – and gain as comprehensive an understanding, as possible, of the field. Starting off with a look into Mathieu Quiniou’s “Advent of Disintermediation”; describing decentralised architecture – as a formula to “solve the problem of certification of the transaction chain, without using a centralised system, to a third party” (Quiniou, Matthieu. Blockchain : The Advent of Disintermediation, John Wiley & Sons, Incorporated, 2019); However, investigation this architecture posed a few interesting points. For example, Quiniou claims that a Blockchain, in this case the “Bitcoin” Blockchain, is “not, as such, designed to store data other than that required for transactions”. Given that in an environment of my software, transactions are required to transport at least some form of data – albeit Quiniou suggests extensions to this rule of thumb are possible; stating that “a text box has been provided”, this being a “Coinbase text” (Quiniou, Matthieu. Blockchain : The Advent of Disintermediation, John Wiley & Sons, Incorporated, 2019).

To further understand how Blockchain can be used as storage, and in data transmission; I enlisted Sandu A’s “Using Blockchain as Secure and Immutable Storage” – a past BCU project, as relevance. Sandhu portrays that files can be uploaded to a Blockchain as if it was typical storage – rather in this case files would abide by a three-pronged protocol; this being a file being uploaded, its status being checked, and file verification being “performed by uploading the transaction document and original file to the blockchain network” (Sandhu, A, 2019). In relation to my interpretation, this would be a constant on-the-fly process – depending on hardware availability, encryption could be difficult to maintain if overly complex.

It's also fundamental to comprehend how each block of a Blockchain is comprised. Sarmah, S writes that “Blockchain has the property of a database except the fact it stores information in the header ad data in the form of a token or cryptocurrency” (Sarmah, S.S., 2018); emphasising how aspects of blocks, such as “header”’s, “hash”’s, and “nonce”’s (i.e. code miner computes to discover) can all combine, if used correctly, to include both a unique identifier and a trace of its file history. Sarmah’s evaluation also provided further information into how well protected a Blockchain is – being described as “resilient to cyber-attacks due to peer-to-peer nature and network would operate when some of the nodes are offline or under security attack” (Sarmah, S.S., 2018).

Cryptography:

In regards to Cryptography, I specialised my research predominantly into how it can affect User Access Control in Local Area Networks -more so because it’s particular environments, like that, I intend on distributing this software into. Likewise, I also branched a similar path into researching how Cryptography can be implemented in mainstream enterprise environments – starting with Del Rio explaining that “DLT arrangements…” (as mentioned earlier) “…can use cryptography for several purposes, such as identity verification and data encryption” (Del Río, C. 2017). Similarly, this also branched into how both symmetric and asymmetric encryption can be used to authenticate data transmission with “digital signatures”, or “private” keys.

This follows into Harrington and Jenson’s discussion, as briefly mentioned, into how Cryptography can be enforced as User Access Control (UAC) in Local Area Networks, in tandem with both symmetric, and asymmetric, encryption methods. Investigating into how "keys" can be utilised to restrict file access – they theorised how this could be coupled with a ledger-based file system to record false entries; which could be spurred into necessary disciplinary investigations if flag trends are perceived in software. An issue discovered though, with significant relevance – in particular – to myself, is that "asymmetric cryptography is too slow…several orders of magnitude too slow" in relation to file transfer, and authentication with the method (Harrington, A. and Jensen, C, 2003, June).

Instead, both advise that in particularly high-demand filing environments – a mechanism employing both symmetric and asymmetric hand-in-hand is more ideal; where a Log system whereby data is transferred symmetrically, and "asymmetric cryptography is reserved for generation and verification of digital signatures" (Harrington, A. and Jensen, C, 2003, June).

Malware:

Malware, a momentous aspect of this project, predominantly concerns the risk of malicious activity to the fundamental anatomy of a Blockchain-like framework being implemented in this project. Although risks associated with cryptographic techniques do exist; I find those directly affecting Blockchain stability itself more significant. A "51% attack", for instance, refers to a situation where "a group of miners control more than 50% of the network's mining hash rate, or computing power" (Ye, Congcong et al. 2018). To put this into perspective, this correlates to either a single user, or a minority of those in a Blockchain controlling over 50% of a Blockchain's computational power – a pertinent issue where "honest miners" are at greater risk of being exploited. Likewise, "the attackers would be unable to prevent new transactions from gaining confirmation" (Ye, Congcong et al. 2018) – in which case if a situation like this were to arise on a Local Area Network, file-editing transactions could be modified, and used to scapegoat others.

Another issue, though not technically a malicious form of file – but rather an activity – is "double spending". Although not entirely commonplace in Blockchain frameworks, more so for decentralised networks; this can cause data redundancy, and files being mistakenly modified due to naming, or metadata issues. Given that I intend to store my ledger on a Blockchain, "one party cannot take control of all transactions in the network" (Aggarwal, S. and Kumar, N. 2021). In retrospect, this infers a pertinent dilemma into how moderation will be organised, as it must be ensured that no particular node group has overwhelming control to framework transactions.

2.3 Summary

All in all, every theme has contributed, to a degree, my line of thinking going into my implementation. I understand a lot more about the anatomy of a Blockchain, a Distributed Ledging Technology, and how a fine line exists in which no one user, or users, can hold control of a Blockchain. I understand Cryptography is somewhat difficult, and requires an unorthodox optimisation of typical asymmetric Cryptography. Likewise, I understand similar competition to my project – but also a few ideas on how to capitalise features left out or weak.

3 Project Design and Methods

3.1 Introduction

Transitioning into a new stage of my project, this section explores my modus operandi into my Design, and further to this - certain methods that will be implemented to achieve my final result. This section has also been segregated into multiple sections, with details ranging from individual methodologies, how they're limited; a set of user requirements, that'll formulate a comprehensive success criteria for me to abide by; with this, a basic design can be constructed. Finally, my success criteria will be evaluated into a set of tests, exploring a majority of planned features – and ensuring my program covers as wide a demographic as comprehensibly as possible with the features provided.

3.2 Methodology

In consideration, I believe a Spiral-like approach to this would be most ideal. In hindsight to objectives I'd defined earlier, particular ones – like adding a GUI – are not essential, and can be prototyped after earlier iterations are extensively tested for full functionality. Other models, like a Waterfall model – aren't particularly ideal to me; although It's coherently presented; Waterfall Model's fallback in that if one section fails, you must revert back in order to continue once more. Waterfall Model's are also more so developed for teams, or those in enterprise; where a team can be distinguished into sections operating on different sections of each task all at once – although in a Spiral configuration this distinguishment isn't easily noticeable, making it far more ideal in the particular use-case is a single individual like myself.

3.3 Limitations and Options

In terms of all themes I've considered above, not all them explicit contained "Methods" as such, given they rather explain both certain concepts, and how their findings can relate into my own project. I don't believe factors, such as cost and availability, to name a couple of instances – matter too much in this case; as modern machines are more than capable of supporting forms of Cryptography – and peer-to-peer nodding in Blockchain, especially in a Local Area Network. All that comes to mind is that if LAN Storage was to be expanded – it has potential to become increasingly expensive if users demand more storage.

3.4 Design Specification/User Requirements

- Network connectivity, connect to Local Area Network with support for real-time symmetric, asymmetric encryption.

- Allow software to be installed on all devices, allow mechanism to constantly scan files for new changes, develop AI model in real-time with outsourced processing to develop malicious activity trends; discover users with malicious intent.

- Ensure hardware is physically capable of modern AES-256 encryption, and SHA-256 hashing.

- Ensure UAC is switched regularly every given time interval, avoids risk of 51% attack, unskewed Blockchain that would severely disrupt security of framework.

- Must have fully-fledged data protection policy in order, ensure it abides with both national and international legislation.

3.5 Testing

- Network Connectivity; test with other computers in Local Area Network, connect to modem/router

- Encryption testing; Establish symmetric key with other node in Local Area Network, provide private digital signature on received encrypted file – test if decrypts.

- Testing AI Model; Input given values from "test" set into certain files, identify whether software has reported it

- UAC Hierarchy; every other day – ensure software asks UAC is being swapped to avoid "51% attack".

- Data-Protection Policy: Integrate data legislation directly into application, ensure administrator has completed entire data policy sheet – if not disable all software features.

3.6 Summary and Conclusions

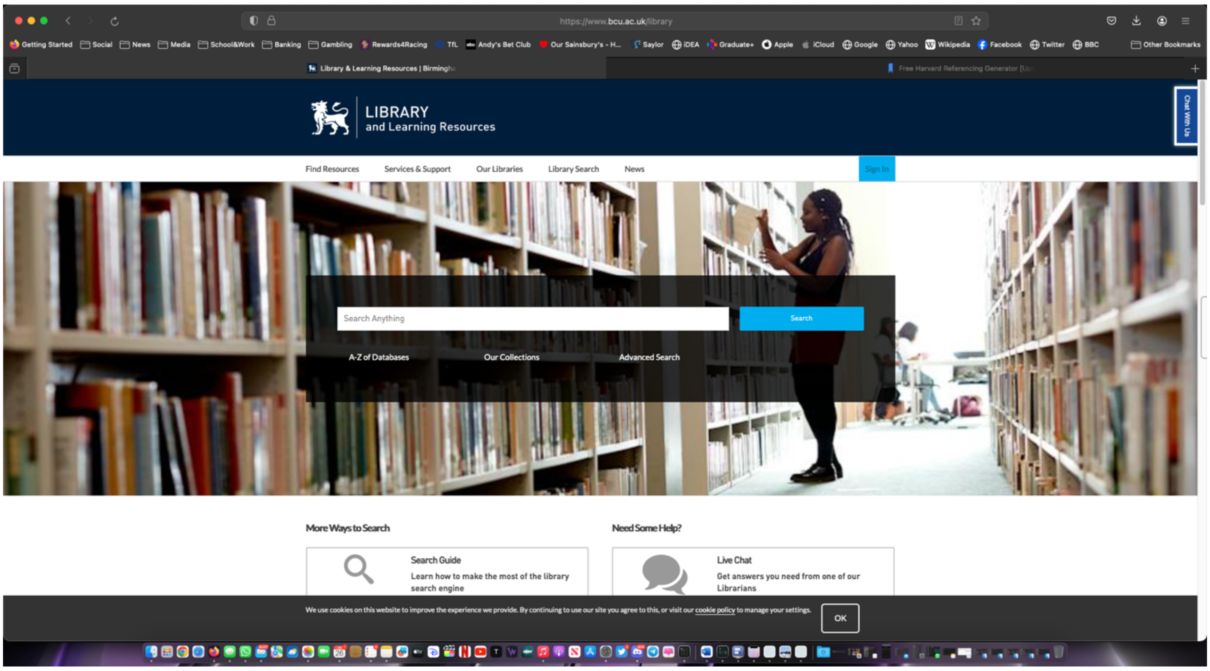
To conclude, I'm going to approach my implementation with a "Spiral"-model mindset; with certain features, such as the likes of a GUI, only being complemented to final prototypes if time permits. Similarly, this is also because of Waterfall's poor solo design in that all features must be completed, in entirety, in linear order; which more often than not – isn't realistic. The main points addressed, as user requirements, entail a machine having sincere cryptographic capabilities; to which a majority of computational hardware this day and age are more than capable of doing just that.

Similarly, all devices within a Local Area Network require my software to be installed, and configured to operate as cryptographically-secure nodes in a Blockchain network – where private keys (i.e. “signatures”) are automatically distributed to support asymmetric local encryption/decryption. To minimise as great a risk of malware as possible, UAC is going to be regularly rotated to reduce the risk of a “51%” attack being initiated, and likewise LAN administrators will be required to import their own policy entirely into my software.

In terms of next steps, I’m going to begin my implementation in its entirety. Now that I have a clear road plan for what both I, and a user, will require from my software; a Model of how to develop it, and an ideal structure in my mind of how it’s to be organised; I can proceed forward with great confidence.

4 Appendix

Appendix One; BCU Library Search:

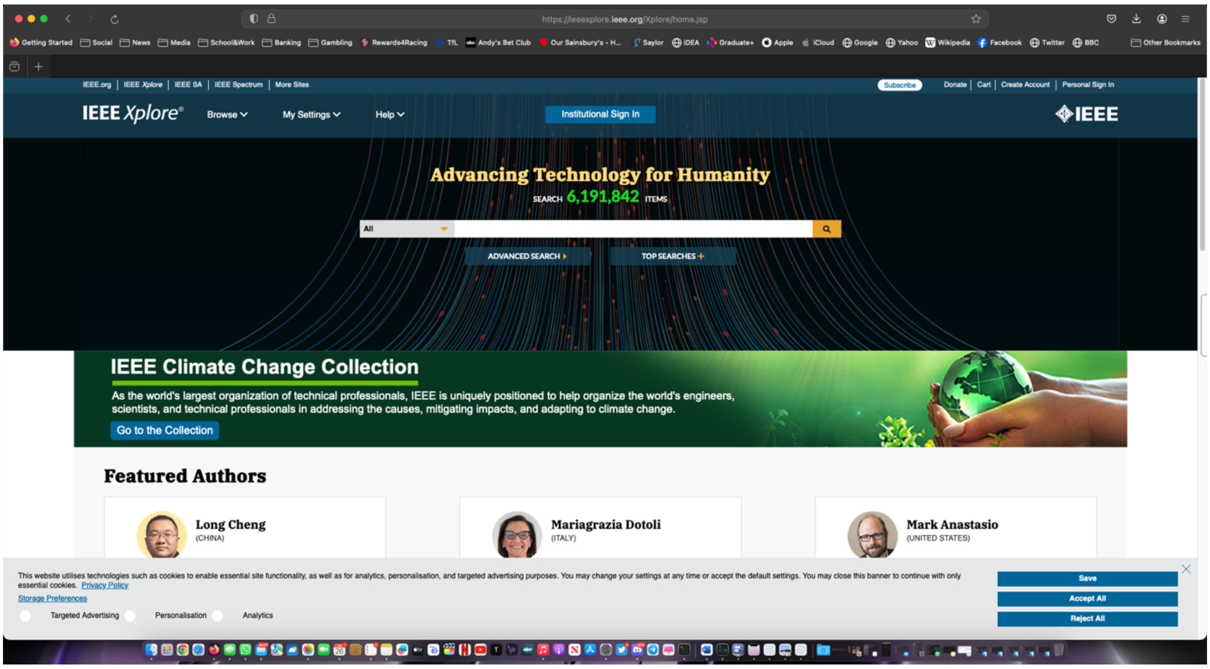


Learn how to make the most of the library search engine

Get answers you need from one of our Librarians

We use cookies on this website to improve the experience we provide. By continuing to use our site you agree to this, or visit our cookie policy to manage your settings.

Appendix Two; IEEE Xplore Home Page:



IEEE Climate Change Collection

As the world's largest organization of technical professionals, IEEE is uniquely positioned to help organize the world's engineers, scientists, and technical professionals in addressing the causes, mitigating impacts, and adapting to climate change.

Featured Authors

Long Cheng (CHINA)

Mariagrazia Dotoli (ITALY)

Mark Anastasio (UNITED STATES)

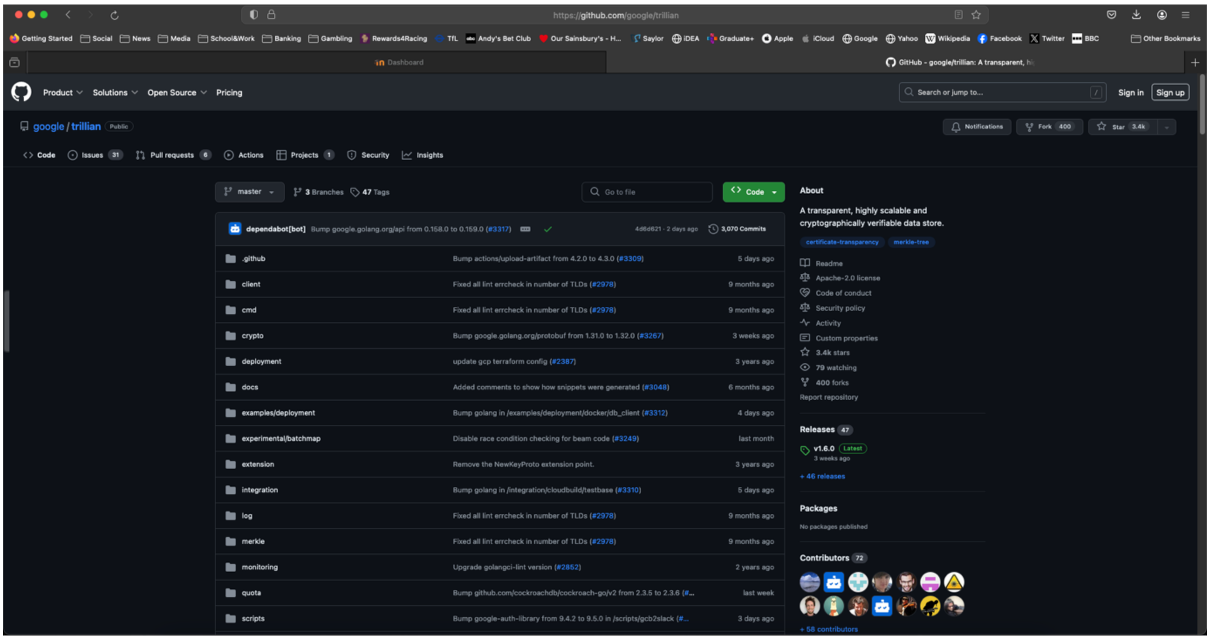
This website utilises technologies such as cookies to enable essential site functionality, as well as for analytics, personalisation, and targeted advertising purposes. You may change your settings at any time or accept the default settings. You may close this banner to continue with only essential cookies. Privacy Policy

Storage Preferences

○ Targeted Advertising ○ Personalisation ○ Analytics

Appendix Three: XMind Theme Mindmap:

Appendix Four: “Trillian” (“transparency.dev”) GitHub repository (https://github.com/google/trillian):



google / trillian Public

dependabot[bot] Bump google.golang.org/api from 0.158.0 to 0.159.0 (#3317) 4d6d621 · 2 days ago 3,070 Commits

.github	Bump actions/upload-artifact from 4.2.0 to 4.3.0 (#3309)	5 days ago
client	Fixed all lint errcheck in number of TLDs (#2978)	9 months ago
cmd	Fixed all lint errcheck in number of TLDs (#2978)	9 months ago
crypto	Bump google.golang.org/protobuf from 1.31.0 to 1.32.0 (#3267)	3 weeks ago
deployment	update gcp terraform config (#2387)	3 years ago
docs	Added comments to show how snippets were generated (#3048)	6 months ago
examples/deployment	Bump golang in /examples/deployment/docker/db_client (#3312)	4 days ago
experimental/batchmap	Disable race condition checking for beam code (#3249)	last month
extension	Remove the NewKeyProto extension point.	3 years ago
integration	Bump golang in /integration/cloudbuild/testbase (#3310)	5 days ago
log	Fixed all lint errcheck in number of TLDs (#2978)	9 months ago
merkle	Fixed all lint errcheck in number of TLDs (#2978)	9 months ago
monitoring	Upgrade golangci-lint version (#2852)	2 years ago
quota	Bump github.com/cockroachdb/cockroach-go/v2 from 2.3.5 to 2.3.6 (#...	last week
scripts	Bump google-auth-library from 9.4.2 to 9.5.0 in /scripts/gcb2slack (#...	3 days ago

About

A transparent, highly scalable and cryptographically verifiable data store.

certificate-transparency merkle-tree

Readme
Apache-2.0 license
Code of conduct
Security policy
Activity
Custom properties
3.4k stars
79 watching
400 forks
Report repository

Releases 47

v1.6.0 Latest
3 weeks ago

+ 46 releases

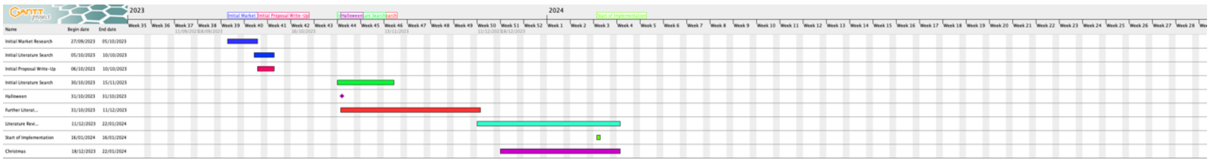
Packages

No packages published

Contributors 72

+ 58 contributors

Gantt Chart:



Name	Begin date	End date
Initial Market Research	27/09/2023	05/10/2023
Initial Literature Search	05/10/2023	10/10/2023
Initial Proposal Write-Up	06/10/2023	10/10/2023
Initial Literature Search	30/10/2023	15/11/2023
Halloween	31/10/2023	31/10/2023
Further Literat...	31/10/2023	11/12/2023
Literature Revi...	11/12/2023	22/01/2024
Start of Implementation	16/01/2024	16/01/2024
Christmas	18/12/2023	22/01/2024

5 References:

Amazon Web Services, Inc. (n.d.). Amazon QLDB. [online] Available at: https://aws.amazon.com/qldb/?c=bl&sec=srv.

Amazon Web Services, Inc. (n.d.). Amazon QLDB Features. [online] Available at: §§https://aws.amazon.com/qldsb/features/?pg=ln&sec=hs.

ransparency.dev. (n.d.). An open-source append only ledger | Trillian. [online] Available at: https://transparency.dev [Accessed 4 Oct. 2023].

Godfrey-Welch, Darlene; Lagrois, Remy; Law, Jared; Anderwald, Russell Scott; and Engels, Daniel W. (2018) "Blockchain in Payment Card Systems," SMU Data Science Review: Vol. 1: No. 1, Article 3.

wiki.hyperledger.org. (n.d.). Hyperledger Fabric - Hyperledger Fabric - Hyperledger Foundation.

Harrington, A. and Jensen, C., 2003, June. Cryptographic access control in a distributed file system. In *Proceedings of the eighth ACM symposium on Access control models and technologies* (pp. 158-165).

Ye, Congcong et al. “Analysis of Security in Blockchain: Case Study in 51%-Attack Detecting.” 2018 5th International Conference on Dependable Systems and Their Applications (DSA). IEEE, 2018. 15–24. Web.

Langsch, V. “Speechless a Virtual Personal Assistant.” Faculty of Computing, Engineering and the Built Environment, 2018. Print.

sanvignesh (2023). *Blockchain vs Database: An In-Depth Comparison*. [online] Medium. Available at: https://medium.com/@vadivelavsbtcrtctech2/blockchain-vs-database-an-in-depth-comparison-2ab0e211827c [Accessed 29 Jan. 2024].

Brilliant.org. (2016). *Merkle Tree | Brilliant Math & Science Wiki*. [online] Available at: https://brilliant.org/wiki/merkle-tree/.

docs.aws.amazon.com. (n.d.). Quotas and limits in Amazon QLDB - Amazon Quantum Ledger Database (Amazon QLDB). [online] Available at: https://docs.aws.amazon.com/qldb/latest/developerguide/limits.html [Accessed 8 Oct. 2023].

Del Río, C. (2017). Use of distributed ledger technology by central banks: A review. Enfoque UTE, 8(5), p.1. doi:https://doi.org/10.29019/enfoqueute.v8n5.175.

Aggarwal, S. and Kumar, N. (2021). Chapter Twenty - Attacks on blockchain☆☆Working model. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0065245820300759

Quiniou, Matthieu. Blockchain : The Advent of Disintermediation, John Wiley & Sons, Incorporated, 2019. ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/bcu/detail.action?docID=5781105

Sandhu, A. “Using Blockchain as Secure and Immutable Storage.” Faculty of Computing,

Engineering and the Built Environment, 2018. Print.

Sarmah, S.S., 2018. Understanding blockchain technology. Computer Science and Engineering, 8(2), pp.23-29.

6 Bibliography

Birmingham City University. (n.d.). Library & Learning Resources. [online] Available at: https://www.bcu.ac.uk/library.

IEEE (n.d.). IEEE Xplore Digital Library. [online] Ieee.org. Available at: https://ieeexplore.ieee.org/Xplore/home.jsp.

Amazon Web Services, Inc. (n.d.). Amazon QLDB. [online] Available at: https://aws.amazon.com/qldb/?c=bl&sec=srv.

Amazon Web Services, Inc. (n.d.). Amazon QLDB Features. [online] Available at: https://aws.amazon.com/qldsb/features/?pg=ln&sec=hs.

ransparency.dev. (n.d.). An open-source append only ledger | Trillian. [online] Available at: https://transparency.dev [Accessed 4 Oct. 2023].

Godfrey-Welch, Darlene; Lagrois, Remy; Law, Jared; Anderwald, Russell Scott; and Engels, Daniel W. (2018) "Blockchain in Payment Card Systems," SMU Data Science Review: Vol. 1: No. 1, Article 3.

wiki.hyperledger.org. (n.d.). Hyperledger Fabric - Hyperledger Fabric - Hyperledger Foundation.

Harrington, A. and Jensen, C., 2003, June. Cryptographic access control in a distributed file system. In *Proceedings of the eighth ACM symposium on Access control models and technologies* (pp. 158-165).

Ye, Congcong et al. “Analysis of Security in Blockchain: Case Study in 51%-Attack Detecting.” 2018 5th International Conference on Dependable Systems and Their Applications (DSA). IEEE, 2018. 15–24. Web.

Xu, J.J. (2016). Are blockchains immune to all malicious attacks? Financial Innovation, 2(1). doi:https://doi.org/10.1186/s40854-016-0046-5.

Langsch, V. “Speechless a Virtual Personal Assistant.” Faculty of Computing, Engineering and the Built Environment, 2018. Print.

sanvignesh (2023). *Blockchain vs Database: An In-Depth Comparison*. [online] Medium. Available at: https://medium.com/@vadivelavsbtcrtctech2/blockchain-vs-database-an-in-depth-comparison-2ab0e211827c [Accessed 29 Jan. 2024].

Budhi, V. (n.d.). *Council Post: Advantages And Disadvantages Of Blockchain Technology*. [online] Forbes. Available at: https://www.forbes.com/sites/forbestechcouncil/2022/10/20/advantages-and-disadvantages-of-blockchain-technology/?sh=2a53bfbb3453 [Accessed 29 Jan. 2024].

transparency.dev. (n.d.). An open-source append only ledger | Trillian. [online] Available at: https://transparency.dev [Accessed 4 Oct. 2023].

GitHub. (2023). Trillian: General Transparency. [online] Available at: https://github.com/google/trillian [Accessed 4 Oct. 2023].

transparency.dev. (n.d.). Trillan helps you reliably log all actions performed on your servers | Trillian. [online] Available at: https://transparency.dev/application/reliably-log-all-actions-performed-on-your-servers/#limitations [Accessed 8 Oct. 2023].

Brilliant.org. (2016). *Merkle Tree | Brilliant Math & Science Wiki*. [online] Available at: https://brilliant.org/wiki/merkle-tree/.

docs.aws.amazon.com. (n.d.). Quotas and limits in Amazon QLDB - Amazon Quantum Ledger Database (Amazon QLDB). [online] Available at: https://docs.aws.amazon.com/qldb/latest/developerguide/limits.html [Accessed 8 Oct. 2023].

Del Río, C. (2017). Use of distributed ledger technology by central banks: A review. Enfoque UTE, 8(5), p.1. doi:https://doi.org/10.29019/enfoqueute.v8n5.175.

Aggarwal, S. and Kumar, N. (2021). Chapter Twenty - Attacks on blockchain☆☆Working model. [online] ScienceDirect. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0065245820300759

Quiniou, Matthieu. Blockchain : The Advent of Disintermediation, John Wiley & Sons, Incorporated, 2019. ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/bcu/detail.action?docID=5781105

Sandhu, A. “Using Blockchain as Secure and Immutable Storage.” Faculty of Computing,

Engineering and the Built Environment, 2018. Print.

Sarmah, S.S., 2018. Understanding blockchain technology. Computer Science and Engineering, 8(2), pp.23-29.