An Investigation into the Feasibility of using Distributed Digital Ledger Technology for Digital Forensics for Industrial IoT

Phillip Fitzpatrick and Christina Thorpe
Technological University Dublin, Ireland

Phillip.fitzpatrick@tudublin.ie
Christina.thorpe@tudublin.ie

Abstract: The domain of Digital Forensics for the Industrial Internet of Things (IIoT) and the proposed use of a Distributed Digital Ledger (DDL), has for the most part been theoretical in nature within the current literature. The work in this paper explores the practical feasibility of using DDL technology for Digital Forensics in the IIOT context. We detail a new methodology for testing the performance of writing to and reading from a DDL in an IIOT environment, and present findings on the overhead associated with storing and retrieving IIoT transactions in a DDL. We conclude that while it is possible to build and use a DDL for storing IIoT transactions, there are limitations to the number of sensors that can be supported by a single implementation and the time it takes to retrieve transactions may be too high to be practical for Digital Forensics.

Keywords: Digital Forensics, IIOT, Blockchain, Distributed Digital Ledge, Performance

1. Introduction

Distributed Ledger Technology (DLT) [15, 7] such as Blockchain has garnered significant interest in recent years due to its potential to provide a secure and transparent means of recording and verifying transactions. This has made it a popular choice for applications in various industries, including the Internet of Things (IoT) [8]. In the context of IoT forensics, blockchain can be used to track and verify the actions and events that take place within a connected device network [12]. One key advantage of using blockchain for IoT forensics is its ability to provide an immutable record of events. This means that once a transaction is recorded on the blockchain, it cannot be altered or deleted, providing a tamper-evident record of events. This can be useful in forensic investigations, as it allows investigators to trace the actions of a connected device and determine nature of blockchain technology allows for increased security and resilience against tampering or unauthorized access. This can help to protect the integrity of forensic data and ensure that it is not compromised during an investigation.

The adoption of public cloud has resulted in the establishment of a new domain called Cloud Forensics, which is the application of Digital Forensics in cloud computing. For IIoT devices, the storage of the data that they emit, given the volume of transactions, will typically result in the data being stored in the Cloud. This presents significant challenges for a forensic investigator, as the data from an IIoT device will traverse many networks and devices before being stored in the Cloud. One of the more prominent challenges is that the data from the IIoT device could be altered at many points along the way.

The focus of this work is to investigate whether a Distributed Digital Ledger (DDL) could be used to securely store all transactions relating to Industrial Internet of Things (IIoT) devices, so that they can be later retrieved for the purpose of conducting a digital cloud forensics investigation. There are acceptable processes for Digital Forensics, which are admissible as evidence in a court of law, but this is not the case for Cloud Forensics. Several frameworks and methods have been proposed in the research regarding how DDLs could be used to secure IIoT transactions, but much of this research is theoretical in nature. Any prior research that has moved beyond the theory, has not demonstrated how a proposed approach could scale to support an IIoT implementation.

To the best of our knowledge and as far as can be ascertained in the literature and in the commercial environment, there is no known approach for such a study. So, to comply with a suitable scientific method, a new methodology - the Industrial Internet of Things Simulated Measurement Methodology (IIoT SM-M) - has been devised and employed in this work.

The aim of this paper is to investigate the feasibility of using DDL technology for Digital Forensics for IIoT. Several research questions that were considered for this are:

1. Can a suitable DDL technology platform be built to support Digital Forensics as it applies to IIoT?
2. What, if any, are the challenges with building such a platform and how could these challenges, if applicable, be overcome?
3. What is the overhead with storing encrypted transactions from IIoT devices, when compared to plaintext transactions?
4. What is the overhead with retrieving encrypted transactions from IIoT devices, when compared to plaintext transactions?

5. What is the highest number of IIoT devices that the platform is able to support?

2. Related work

The literature review is aimed at understanding the field of Digital Forensics with a specific focus on its application within the Industrial Internet of Things (IIoT). It critically examines three significant survey papers that provide a comprehensive state-of-the-art review of Digital Forensics in the context of the Internet of Things [2], [13], [5]. The scope of Digital Forensics is extensive, encompassing areas such as disk, mobile, network, wireless, and cloud forensics [6], with particular attention to Cloud Forensics models tailored for IoT and IIoT that utilize Distributed Ledger Technology (DDL).

In terms of Cloud Forensics, it is recognized to include essential elements like the IoT device, the DDL, and the Cloud environment [11]. The focus of this research is on piecing together evidence from these components to create a chronological series of events for forensic analysis.

The discussion extends to several theoretical frameworks and models proposed for storing IoT device data on DDLs, which ensures the integrity of the data through encrypted transactions. Among these, some frameworks have been partially tested using Ethereum, highlighting the secure, decentralized nature of such digital transaction ledgers [17]. However, despite these developments, many models, including a blockchain-based framework for storing IoT communications as transactions [14], and the FIF-IoT framework [10], remain largely theoretical and have not demonstrated scalability or practical implementation in real-world settings.

Additional proposals explored include using DDLs for creating tamper-resistant solutions for audit trails in IoT systems [16], which could later assist in digital forensics. Another notable initiative is the BPIIoT platform for IIoT, which utilizes blockchain technology to enable cloud-based manufacturing. This platform was demonstrated using a single machine prototype, but questions about its scalability remain unanswered [4]. Similarly, the BLOFF IoT forensics model aims to ensure the integrity of forensic evidence but has not been implemented beyond theoretical discussions [1].

Each of these initiatives reflects a growing interest in integrating blockchain and DDL technologies with IoT systems to enhance the security and verifiability of data, crucial for effective forensic investigations in the realm of IIoT. However, the literature indicates a gap in real-world application and evidence of scalability, pointing to areas needing further research and development.

Table 1: Models and Frameworks from the Literature Reviewed.

Models and Frameworks				
Name	Date	Experiment	Ledger	Scaled
	2019	Yes	Ethereum	No
FIF-IoT	2018	Yes	Ethereum	No
Probe-IoT	2018	No	None	No
	2018	No	None	No
BPIIoT	2016	Yes	Ethereum	No
BLOFF	2021	No	None	No

3. Methodology

In order to confirm the aim of this paper, a quantitative research approach needed to be undertaken. To achieve this, two applications were developed using the Python programming language and the integrated development environment for Microsoft Windows, Microsoft Visual Studio. A real-world IIoT transactions dataset, comprising thermostat temperature values, was used to simulate reading IIoT transactions from thermostat sensors and a simulation measurement methodology was devised.

3.1 IIoT SM-Methodology

The Industrial Internet of Things Simulation Measurement Method- ology that was devised for this work contains 5 stages: Platform Installation and configuration, Platform Initialisation, Use Case Experiment Execution, Results Recording, and Results Analysis. Each of these stages are described in relevant subsections below.

3.2 Python Applications

To carry out the quantitative research for this paper, two Python applications were developed (these will be made publicly available if accepted for publication.). One application was used to read and write IIoT transactions from a dataset and process them using the PostgreSQL database, while the other application was used to read and write the same IIoT transactions and process them for the Iroha digital ledger.

3.3 Datasets

For an IIoT implementation, there would be hundreds and, in some cases, thousands of sensors that would be emitting readings every second, 24 hours per day. By way of a simple calculation, for a large manufacturing operation with over 1,000 sensors transmitting every second, over 1.4 million transactions would be emitted in any one twenty-four-hour period. While the size of transactions from such a sensor would be relatively small, it is the volume of transactions needing to be written or read that presents a challenge, particularly as it relates to IIoT and Cloud Forensics. For this paper, several data repositories of different datasets were reviewed, and a suitable data set was identified to execute the experiments. The dataset chosen contains over 400,000 thermostat temperature readings, collected from two building thermostat sensors, reporting every second, over a two-year period (https://tinyurl.com/4d2tsnmt). The dataset was created by the Intelligent Security Group UNSW Canberra, Australia in support of Industry 4.0, [3].

3.4 Use Cases

Each experiment measured the time taken to write or read an IIoT transaction into or from the PostgreSQL database or the Iroha digital ledger. Three use cases were evaluated:

- Use Case 1: Writing or reading an unencrypted IIoT transaction directly to a PostgreSQL database table.
- Use Case 2: Writing or reading an unsigned IIoT transaction into the Iroha digital ledger.
- Use Case 3: Writing or reading a signed IIoT transaction into the Iroha digital ledger.

The experiments were conducted five times for each use case, and the average time was recorded as the result. The types of IIoT transactions tested were:

- Unencrypted IIoT transaction: Directly written or read from the PostgreSQL database without encryption.
- Unsigned IIoT transaction: Written or read from the Iroha digital ledger without a signature.
- Signed IIoT transaction: Written or read from the Iroha digital ledger with a signature, implying encryption.

The results were collated and analysed to identify challenges associated with each use case.

3.5 Database Performance Monitoring

The monitoring of PostgreSQL databases during each experiment was conducted using pgAdmin 4, which features a performance monitoring dashboard. This dashboard provides real-time updates on the health and performance of the database, displaying various statistics such as the number of active database sessions, transactions per second (including commits and rollbacks), tuples processed (inserted, updated, deleted), tuples fetched, and block I/O operations (blocks read from the file system or buffer cache). Key metrics of interest included the transactions per second, which were correlated with the time taken to execute the use cases.

3.6 Iroha Digital Ledger

The DDLs reviewed for this work are hosted by the Hyperledger Foundation, who "support the development of enterprise-grade, cross-industry open platforms for distributed ledgers" (https://www.hyperledger.org/about/join). They include BESU, FABRIC, INDY, IROHA, and SAWTOOTH.

IROHA has been selected for this paper due to its advantageous features for IoT projects requiring distributed ledger technology. Firstly, its simplicity and ease of integration make it highly suitable for such applications. Secondly, IROHA necessitates the use of either a PostgreSQL database or RocksDB as its underlying data repository, aligning with the paper's objectives to analyze and compare read and write times between a DDL and a database. Lastly, IROHA supports a Python library, which facilitates the development of applications that interact with the DDL, further enhancing its utility for this research.

The Iroha digital ledger has a limitation on the type of values that it can store. An Asset, a defined type within the Iroha digital ledger, can only contain a value, which can only be increased or decreased. This presented a challenge for this paper, since value types other than numeric needed to be stored. To overcome this limitation, a User Object was used to store the IIoT transactions, since it is possible to set a value on the User Object of up to 2,048 bytes in size. In the Iroha platform, the setting of this value is achieved in exactly the same way as adding a transaction to the digital ledger. The transaction is signed with a private key and is then written to the Iroha digital ledger.

3.7 Testbed

For this research, a standalone project platform was set up using both commercially available and open-source hardware and software to conduct experiments. The physical infrastructure comprises a single HP ProLiant DL380 Gen10 Intel server running Microsoft Windows Server 2020 R2 and configured with Microsoft Windows

Hyper-V. The server features two Intel Xeon Gold 6130 CPUs at 2.10GHz with 16 cores and 32 logical processors each, 384MB of physical memory, 1.92 TB SSD in RAID 1 for OS internal storage, and 27.8 TB of external storage on an HPE Smart Array P816i-a SR. The logical setup, depicted in Figure 1, includes two virtual instances on Hyper-V: one runs a Microsoft Windows 10 environment with Microsoft Visual Studio (16 virtual processors, 128 MB memory, 1 TB storage), and the other an Ubuntu environment equipped with a Docker setup hosting three containers (12 virtual processors, 128 MB memory, 1 TB storage).

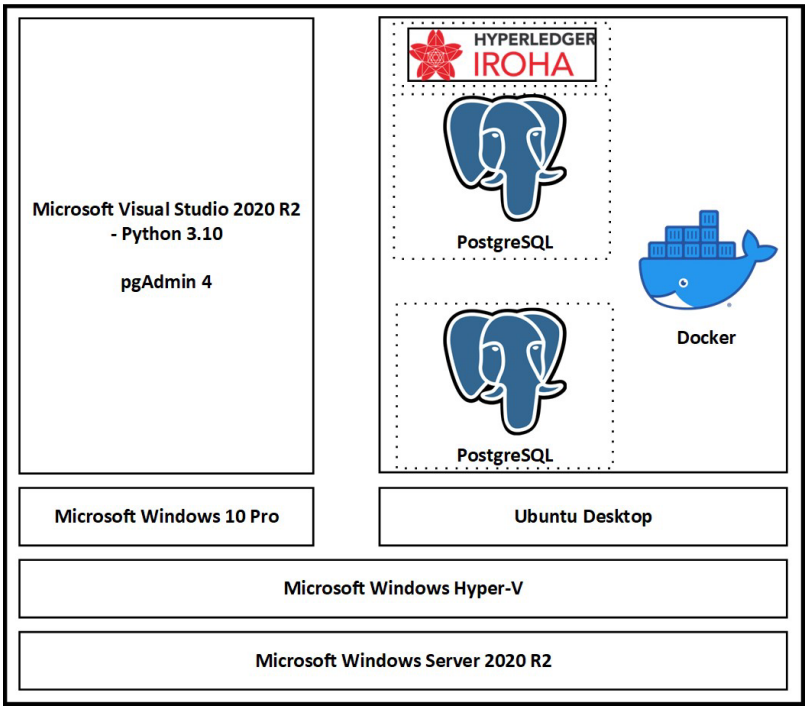

Figure 1: Logical Architecture

The research setup involved deploying various software on an HP ProLiant DL380 Gen10 server. Microsoft Windows Server 2019 Datacenter edition and Microsoft Windows Server 2020 R2 with Hyper-V were installed on the server. Within Hyper-V, Ubuntu Desktop 20.04 was set up, hosting Docker version 20.10.12, which in turn managed three containers. Hyperledger Iroha, utilizing PostgreSQL as its database for storing encrypted IIoT transactions, was deployed on one of these Ubuntu Docker containers. Additionally, two separate PostgreSQL database instances were configured in individual Docker containers—one for Iroha's object store and another for unencrypted IIoT transactions.

On the software development front, Microsoft Windows 10 Pro was installed in another Hyper-V virtual environment, where the Community Edition of Visual Studio and Python 3.10 were set up. This setup was used for programming and running experiments. The psycopg2 and Iroha Python libraries were installed to support database interactions and blockchain functions, respectively. Lastly, pgAdmin 4 version 6.8 was installed on Windows 10 for database creation, configuration, and monitoring.

4. Experiments and results

To achieve the aim of investigating the feasibility of using Distributed Digital Ledger technology for Digital Forensics for Industrial Internet of Things, six experiments were performed on the experimental platform developed for this work. This section details the series of experiments that were executed and records how long it took to write or read unencrypted or encrypted IIoT transactions into or from a PostgreSQL database or the Iroha digital ledger. The IIoT transactions were of various volumes ranging from 100k to over 400k and of varying sizes, ranging from 1k to 4k.

4.1 Experiment Initialisation

Before executing any of the experiments, a series of experiment initialisation steps were performed.

1. **Stop** the Docker containers containing the PostgreSQL databases and the Iroha digital ledger.
2. **Delete** the Docker containers containing the PostgreSQL databases and the Iroha digital ledger.
3. **Run** the Docker container containing the standalone PostgreSQL database.

4. **Run** the Docker container containing the PostgreSQL database for the Iroha digital ledger.
5. **Run** the Docker container containing the Iroha digital ledger.
6. **Confirm** that all containers are up and running.

Performing the above steps before executing any of the experiments ensured that the project platform was clean and stable, that nothing was left behind from a previous experiment and that there was no impact between experiments.

Table 2: Experiment One - Writing, in seconds, 1,000 IIoT Transactions Time.

Use Case	R1	R2	R3	R4	R5	Avg
1: Unencrypted	0.9		0.94	0.99	1.04	1.25	1.02
2: Unsigned	0.48		0.48	0.44	0.46	0.49	0.47
3: Signed & Encrypted	5.63		5.78	5.56	5.79	5.21	5.59

4.2 Ex1 - Writing 1,000 IIoT Transactions

In an experiment with a baseline of 100 IIoT transactions, a noticeable time difference was observed when writing signed transactions to the Iroha digital ledger. To explore this further, the number of transactions was increased to 1,000, and the experiment involved writing these transactions to both the PostgreSQL database and the Iroha digital ledger. The write times were recorded, and database performance was continuously monitored, showing no anomalies.

Three specific use cases were tested:

- Use Case 1: Unencrypted Transactions - A file with 1,000 IIoT thermostat transactions was processed and written to PostgreSQL. The process was repeated five times to confirm consistent results.
- Use Case 2: Unsigned Transactions - The same file was processed and written unsigned to the Iroha ledger, and similarly repeated for reliability.
- Use Case 3: Signed Transactions - The file was processed, and transactions were signed (thus encrypted) before writing to Iroha, with the process also repeated multiple times.

The results indicated a minor time difference between writing to PostgreSQL and unsigned writing to Iroha. Notably, signed transactions took significantly longer, consistent with the overhead of encryption via private key signing. In a real-world scenario, it is likely that there would be up to a one second time delay between each transaction, but if there were 100 or even 1,000 IIoT devices, all trying to write to the Iroha ledger at the same time, this could become a performance issue for the system. Experiment 4.5 will look at a simulation of 10 IIoT devices, all trying to write 1000 IIoT transactions at the same time to evaluate how the system performs. Experiment 4.5 will simulate writing by 10 IIoT devices simultaneously to assess system performance, exploring potential delays in a real-world scenario.

4.3 Ex2 - Reading 1,000 IIoT Transactions

Being able to read transactions from the Iroha digital ledger is an important requirement in the analysis phase of Cloud Forensics. To investigate this, an experiment was executed to read 1,000 IIoT transactions directly from the PostgreSQL database and from the Iroha digital ledger. The time taken to read the 1,000 IIoT transactions was recorded. For this experiment, there are only two use cases since it is looking to determine whether there may be an overhead when retrieving the signed transactions. The database performance was monitored throughout all experiments and no anomalies were observed. The results are shown in Table 3.

Table 3: Experiment Two - Reading, in seconds, 1,000 IIoT Transactions Time

Use Case	R1	R2	R3	R4	R5	Avg
1: Unencrypted	0.79	0.71	0.74	0.76	0.72	0.74
2: Signed & Encrypted	7.76	7.26	7.61	7.27	7.28	7.44

- Use Case 1: Unencrypted Transactions: A single IIoT transaction from a set of 1,000 thermostat readings was retrieved 1,000 times from the PostgreSQL database. The total processing time was recorded, and the test repeated four times to ensure reliable results, as shown in Figure 3. Throughout the experiment, server performance was monitored via the pgAdmin 4 dashboard to maintain database consistency.

- Use Case 2: Signed Transactions: Similarly, a single IIoT transaction from the Iroha digital ledger was retrieved 1,000 times. The process mirrored Use Case 1, with the total time taken recorded and the test repeated four times for statistical reliability.

The experiment confirmed that both use cases could successfully retrieve transactions from the respective databases. Table 3 details the average time required to retrieve one IIoT transaction 1,000 times for both use cases. There was a noticeable time difference between the two databases; it took just over 1 second in Use Case 1 and more than 7 seconds in Use Case 2 to complete the reads. This time discrepancy is expected due to the decryption required for each transaction in the Iroha ledger. Despite the added time, there were no significant performance issues, suggesting that the Iroha digital ledger could handle the decryption workload without major delays in a real-world scenario.

4.4 Ex3 - Writing varying IIoT Transaction Sizes

The purpose of experiment three was to investigate whether there would be an impact to the overall performance of the Iroha digital ledger for different IIoT transaction sizes. For this experiment, three file sizes, 1k, 2k and 4k were used as the IIoT transaction sizes. The experiment was executed by processing 1,000 IIoT transactions of the three different file sizes and writing them directly into the PostgreSQL database and into the Iroha digital ledger. The time taken to process the 1,000 IIoT transactions was recorded. The database performance for writing the 4k IIoT transaction size was monitored throughout all experiments and no anomalies were observed. The results are shown in Table 4.

Table 4: Experiment Three - Writing, in seconds, different file size IIoT Transactions Time.

Use Case	Size	R1	R2	R3	R4	R5	Avg
1: Unencrypted	1KB	0.95	1.07	0.98	0.88	0.91	0.96
	2KB	0.96	1.04	1	0.99	1.06	1.01
	4KB	0.94	1.09	1.15	1.01	1.02	1.04
2. Unsigned	1KB	0.6	0.56	0.63	0.53	0.57	0.58
	2KB	0.71	0.7	0.72	0.73	0.75	0.72
	4KB	0.99	0.89	0.93	0.89	0.94	0.93
3: S & E	1KB	5.56	5.64	5.74	5.71	5.72	5.69
	2KB	5.78	5.68	5.74	5.65	5.74	5.72
	4KB	6.13	6	5.96	6.09	5.96	6.03

- Use Case 1: Unencrypted Transactions: Files of 1k, 2k, and 4k, each containing a single IIoT transaction, were processed 1,000 times into the PostgreSQL database. The total processing time was recorded and repeated four times for accuracy, with server performance consistently monitored via the pgAdmin 4 dashboard.
- Use Case 2: Unsigned Transactions: The same files were processed 1,000 times into the Iroha digital ledger without signing the transactions. The procedure mirrored Use Case 1, including four repetitions to ensure statistical reliability, with results shown in Table 4.
- Use Case 3: Signed Transactions: The transactions were encrypted through signing and processed similarly in the Iroha ledger, with results also documented in Table 4 after four repetitions.

The experiment validated that all three use cases effectively wrote transactions to both the PostgreSQL and Iroha ledgers, as recorded in Table 4. Time to write transactions was slightly faster in the PostgreSQL database and for unsigned transactions in the Iroha ledger, typically under 1 second. Signed transactions in Iroha took about 6 seconds, regardless of file size. The file size (1k, 2k, or 4k) had minimal impact on processing times across all scenarios, indicating that transaction size would likely not pose significant performance issues in real-world applications.

4.5 Ex4 - IIoT device with Multiprocessing

The purpose of this experiment was to simulate ten IIoT sensors, all writing to the Iroha digital ledger at the same time. This was achieved by using the multiprocessing capabilities within Python and spawning multiple processes to execute in parallel. In experiment one, Section 4.2, the time taken to process 1,000 transactions was recorded. For this experiment, the time taken to process 10,000 transactions was recorded. The experiment then involved comparing the time taken to process 10,000 transactions, with ten parallel processes of 1,000 transactions. The experiment was executed by writing the 10,000 IIoT transactions directly into the PostgreSQL database and into the Iroha digital ledger. The time taken to write the IIoT transactions was recorded. The results are shown in Table 5.

Table 5: Experiment Four - 10 IIoT Sensor Simulation Times. Mode M = Multi and Mode S = Single

Use Case		R1	R2	R3	R4	R5	Avg
1: Unencrypted	S	8.88	8.79	8.16	8.74	8.72	8.66
	M	1.18	1.23	1.23	1.23	1.12	1.20
2. Unsigned	S	4.65	4.59	4.7	4.59	4.59	4.62
	M	0.59	0.68	0.57	0.65	0.55	0.61
3: Signed &	S	55.43	55.11	56.71	55.82	55.55	55.72
Encrypted	M	8.51	7.77	9.05	7.74	8.97	8.41

- Use Case 1: Unencrypted Transactions: Two files containing 10,000 and 1,000 IIoT thermostat transactions were processed into PostgreSQL both individually and as 10 parallel processes. Each round was repeated four times for accuracy, with consistent server performance monitoring via pgAdmin 4 showing no anomalies.
- Use Case 2: Unsigned Transactions: Identical files were processed into the Iroha digital ledger without signatures, also as 10 parallel processes. The processing time was recorded over four repetitions, mirroring the first use case with consistent server performance.
- Use Case 3: Signed Transactions: The same files were processed into Iroha with transactions signed and encrypted, executed as 10 parallel processes. This was repeated four times, with consistent server monitoring and performance.

The experiments confirmed effective transaction writing to both PostgreSQL and Iroha across all use cases, as documented in Table 5. Processing 10,000 IIoT transactions showed a near seven-fold reduction in time when executed in parallel. While there was an expected increase in time for processing signed transactions compared to unsigned, this aligned with previous findings and did not present unexpected anomalies.

4.6 Ex5 - Writing/Reading 400k Transactions

For an IIoT implementation, there will be hundreds, if not thousands, of sensors that would each be sending a transaction at very regular intervals. In some cases, this could be as frequent as every second. In the case of a smart city, there would be tens of thousands of sensors. This experiment was executed to write and read over four hundred thousand transactions into and from the Iroha digital ledger. The time taken to execute the writing and reading was recorded. Since the experiment is only looking to determine whether there is an overhead when retrieving the signed transactions, there are only two use cases. The transactions represent readings from just two IIoT sensors over a two-year period. The results are shown in Table 6.

Table 6: Experiment Five- Writing / Reading, in minutes, 400k IIoT Transactions Time.

Use Case	R1	R2	R3	R4	R5	Avg
1. Writing	41.15	41.1	41.2	41.45	41.2	41.22
2. Reading	43.5	43.3	43.78	43.45	43.35	43.48

- Use Case 1: Written Signed Transactions: A file with over four hundred thousand IIoT thermostat transactions was processed into the Iroha digital ledger, line by line. The processing time was recorded and repeated four times for reliability.
- Use Case 2: Read Signed Transactions: One of the 400,000 IIoT transactions written to Iroha was retrieved 400,000 times. The procedure was identical to Use Case 1, with the time recorded and repeated four times for consistency.

Both use cases successfully wrote to and read from the Iroha ledger, with the results displayed in Table 6 showing the average processing time in minutes for the 400k IIoT transactions. Writing took over 40 minutes, while reading took just under 44 minutes, consistent with previous experiments.

For Cloud Digital Forensics, analysing large numbers of transactions like those from an IIoT sensor—which could generate 1,440 transactions daily—suggests that the Iroha ledger can handle substantial loads without significant performance issues. For instance, processing nearly 13 million transactions from 100 sensors over three months would take about four hours if done serially. Future improvements might involve scaling the Iroha system horizontally to enhance efficiency.

5. Key experimentation findings

A total of six experiments have been successfully executed in order to validate the aim of this research. The key findings are as follows:

1. There is more than a five-fold overhead when comparing the time taken to write an IIoT transaction into the DDL that is encrypted, compared to one that is not.
2. There is almost a ten-fold overhead when comparing the time taken to read an IIoT transaction from the DDL that is encrypted, compared to one that is not.
3. When comparing the time taken to write IIoT transactions of different sizes, allowing for the time difference between encrypted and unencrypted IIoT transactions, the difference is minimal.
4. The number of encrypted transactions that can be written to the platform per second is limited to 170. If the number of IIoT devices emitting data every second was to exceed 170, then a platform backlog would occur.
5. When the number of IIoT transactions was increased to 400k, then the number of supported devices was reduced from 170 to 161.

5.1 Outcomes

The research questions that were considered for this paper and the accompanying answers are:

Q1: This paper has successfully demonstrated that it is possible to build a suitable DDL technology platform for the storage and retrieval of encrypted IIoT transactions, for the purpose of conducting a forensic analysis.

Q2: The methodology section describes the limitation with the Hyperledger Iroha and how this limitation was overcome.

Q3: The execution of experiments demonstrated that there is an overhead when storing encrypted IIoT transactions in the platform. Specifically, there is in excess of a five-fold increase in the time difference for storing transactions, which results in a reduction of processing volume to under 200 transactions per second.

Q4: The execution of experiments demonstrated that there is an overhead when retrieving encrypted IIoT transactions from the platform. Specifically, there is almost a ten-fold increase in the time difference for retrieving transactions which results in a reduction of processing volume to under 135 transactions per second.

Q5: The execution of experiments demonstrated that if the number of IIoT devices was to exceed 170, which is the maximum number of transactions that can be processed per second, then a platform backlog would occur.

6. Conclusions

The aim of this research was to investigate the feasibility of using Distributed Digital Ledger technology for Digital Forensics for Industrial Internet of Things.

The outcome of this work has shown that previously suggested methods and frameworks can be successfully implemented using DDL technology for the storage of encrypted IIoT transactions. These transactions can later be retrieved for the purpose of conducting a forensic investigation. The experimental platform built for this paper and the devised simulated measurement methodology, have proven that it is possible to build and use a DDL, like Hyperledger Iroha, for the purpose of storing IIoT transactions, within certain parameters. Specifically, there is a limit to the number of sensors that could be supported by a single implementation and the time it takes to retrieve IIoT transactions may be too high to be practical for the purpose of Digital Forensics.

References

Promise Agbedanu and Anca Delia Jurcut. “BLOFF: a blockchain-based forensic model in IoT”. In: Revolutionary Applications of Blockchain-Enabled Privacy and Access Control. IGI Global, 2021, pp. 59–73.

Ahmed Alenezi et al. “IoT forensics: A state-of-the-art review, challenges and future directions”. en. In: Proceedings of the 4th International Conference on Complexity, Future Information Systems and Risk. Heraklion, Crete, Greece: SCITEPRESS - Science and Technology Publications, May 2019.

Abdullah Alsaedi et al. “TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems”. In: IEEE Access 8 (2020), pp. 165130–165150.

Arshdeep Bahga and Vijay K Madisetti. “Blockchain platform for industrial internet of things”. In: Journal of Software Engineering and Applications 9.10 (2016), pp. 533–546.

Arafat Al-Dhaqm et al. “Digital Forensics Subdomains: The State of the Art and Future Directions”. In: IEEE Access 9 (2021), pp. 152476–152502.

Arafat Al-Dhaqm et al. “Digital Forensics Subdomains: The State of the art and Future Directions”. In: IEEE Access (2021).

Nabil El Ioini and Claus Pahl. “A review of distributed ledger technologies”. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer. 2018, pp. 277–288.
Bahar Farahani, Farshad Firouzi, and Markus Luecking. “The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions”. In: Journal of Network and Computer Applications 177 (2021), p. 102936.
Mahmud Hossain, Yasser Karim, and Ragib Hasan. “FIF-IoT: A forensic investigation framework for IoT using a public digital ledger”. In: 2018 IEEE International Congress on Internet of Things (ICIOT). IEEE. 2018, pp. 33–40.
Md Mahmud Hossain, Ragib Hasan, and Shams Zawoad. “Probe-IoT: A public digital ledger based forensic investigation framework for IoT.” In: INFOCOM workshops. 2018, pp. 1–2.
Jianwei Hou et al. “A survey on digital forensics in Internet of Things”. In: IEEE Internet of Things Journal 7.1 (2019), pp. 1–15.
Randa Kamal, Ezz El-Din Hemdan, and Nawal El-Fishway. “A review study on blockchain-based IoT security and forensics”. In: Multimedia Tools and Applications (2021), pp. 1–32.
Pantaleon Lutta et al. “The complexity of internet of things forensics: A stateof-the-art review”. In: Forensic Science International: Digital Investigation 38 (Sept. 2021), p. 301210.
Jung Hyun Ryu et al. “A blockchain-based decentralized efficient investigation framework for IoT digital forensics”. In: The Journal of Supercomputing 75.8 (2019), pp. 4372–4387.
Ali Sunyaev. “Distributed ledger technology”. In: Internet Computing. Springer, 2020, pp. 265–299.
Magnus Westerlund, Mats Neovius, and Göran Pulkkis. “Providing tamperresistant audit trails with distributed ledger-based solutions for forensics of IOT systems using cloud resources”. In: International Journal on Advances in Security 11.3 & 4 (2018).
Gavin Wood et al. “Ethereum: A secure decentralised generalised transaction ledger”. In: Ethereum project yellow paper 151.2014 (2014), pp. 1–32.