Engineering Blockchain-Based Narrowband Internet of Things Applications for Energy Optimization*

Hafizullah Kakar[1], Vamshi S. Mohan[3], Swapnoneel Roy[1], Ayan Dutta[1], O. Patrick Kreidl[1], Ladislau Bölöni[2], and Sriram Sankaran[3]

[1] University of North Florida, Jacksonville, FL, USA
[2] University of Central Florida, Orlando, FL, USA
[3] Amrita Vishwa Vidyapeetham, Amritapuri, India

Abstract. We optimize the overall energy consumption of a Narrowband Internet of Things (NB-IoT) application created using a hybrid blockchain framework. We accomplish this by engineering the underlying hash function (SHA-256) that is used in different procedures (Unique ID generation, Device Join, and Device Transaction) of the blockchain-based NB-IoT system. In order to reduce the complexity of hash verification, IoT devices in the NB-IoT application are built to save the hashes of their authorized transactions as a linear hash chain rather than the entire Merkle tree. Furthermore, base station memory is dynamically partitioned to improve memory usage efficiency and scalability. Compared with the state-of-the-art approach, our approach considerably reduces the total energy consumption of the state-of-the-art application.

1 Introduction

We investigate how much further energy efficiency may be achieved by engineering the underlying hash function (SHA256), which is a key component of the hybrid blockchain-based model for NB-IoT of Mohan et al. [23]. Using an energy-saving algorithm engineering method, our solution is based on applying an Energy Complexity Model (ECM) proposed by Roy et al. [16], on the SHA256 hash algorithm, which is essential to Mohan et al. [23] hybrid blockchain-based model for NB-IoT.

We experiment with both the standard and energy-reduced implementations of Mohan et al.'s model for NB-IoT for input sizes (in bytes) that are typically found within NB-IoT applications using pyRAPL, a Python package to measure an executable's Runtime Average Power Limit. Our findings demonstrate further savings in energy usage on top of the state-of-the-art hybrid blockchain-based model for NB-IoT by Mohan et al. [23].

Currently, the idea that the lower energy consumption in the NB-IoT module itself translates into a similar reduction in an application that utilizes an entire

* H. Kakar, S. Roy, A. Dutta, O.P. Kreidl, and L. Bölöni acknowledge funding through NSF Grants #1932300 & #1931767.

real-world system is still theoretical. However, our work is the first that we are aware that addresses the optimization of energy of NB-IoT through the technical implementation of one of its component algorithms (SHA256). Furthermore, other essential components of a protected system can also benefit from the suggested energy-saving method, which could lead to even *greener* secured application systems than those suggested by the NB-IoT results yet.
This work expands on the work of Mohan et al. [23]. While experiments with the deployment of a hybrid blockchain in NB-IoT, this work experiments on the energy efficiency of the resulting model for NB-IoT, integrating an energy-optimized version of SHA256, called ECM-SHA256, in the model of Mohan et al. [23].

1.1 Our Contributions

To summarize our contributions:

1. We optimize the energy consumption of the hybrid blockchain-based model NB-IoT by Mohan et al. [23].
2. We achieve this using an *algorithmic* approach, wherein we engineer the underlying SHA256 hash of the system for energy optimization.
3. To the best of our knowledge, ours is the first work to tackle the problem of energy optimization in NB-IoT using an algorithmic engineering approach.

1.2 Related work

The Low Power Wide Area Network (LPWAN) known as the Narrowband Internet of Things (NB-IoT) was created by 3GPP and offers increased coverage, high connection density, extended battery life, and other benefits [6,1]. It uses Single-Carrier FDMA (SC-FDMA) for Uplink communication and Orthogonal Frequency Division Multiplexing (OFDMA) for Downlink communication on a decreased bandwidth of 180 kHz per carrier. NB-IoT does not provide seamless device handover between base stations or low-latency applications. As a result, it works perfectly in low-power stationary Internet of Things devices that communicate little amounts of data irregularly, including smoke detectors and water meters.
Many investigations have been conducted to optimize ON-OFF periods and eDRX (Extended Discontinuous Reception) and PSM (Power Saving Mode) to extend the battery life of NB-IoT devices [21]. Although NB-IoT is already used in stationary applications, experts see it being used in mobile applications in the future, including tracking of pets and sharing of public bikes [18]. This kind of research could lead to the development of mobile applications NB-IoT that are more scalable, have wider coverage, and use less energy [14].
Devices are not automatically assigned to visited base stations to enable continuous smooth services, since NB-IoT does not support authentication and certificate transfer while traversing between different cells [1]. As a result, the devices must manually select base stations. Due to the decreased bandwidth, this could

result in phony base station connections and Denial of Service (DoS) assaults, which would compromise user privacy and data. In this research, a centralized architecture has been suggested to connect devices to cloud servers hosted by the service provider [5]. Device authentication, smooth handover, and device communication requests are handled by cloud servers. However, during periods of high traffic, cloud servers become a bottleneck in handling device queries, causing responses to be delayed. Additionally, when the number of devices and transactions rises, a centralized architecture (e.g. [5]) prevents the system from scaling.

In order to address the aforementioned issues, Mohan et al. [23] has designed a hybrid blockchain architecture to protect NB-IoT and minimize the complexity of hash verification by storing the hashes of authorized transactions as a linear hash chain rather than the whole Merkle tree. This approach uses correspondingly 80.50%, 74.73%, and 50% less memory, processing power, and execution time. To the best of our knowledge, they are the first to suggest a method for effectively allocating memory in resource-constrained mobile NB-IoT applications by splitting up base station memory and storing device hashes separately as linear hash chains rather than Merkle trees.

Quite a lot of research has been done on the subject of energy efficiency in computation. Some perspectives on this include those on hardware-specific platforms, operating systems, hypervisors, and containers [24]; software development and security [8]; fog computing-based platform [22] and algorithms [16,17] [13]. Sometimes, uniquely instrumented equipment is used to obtain energy measurements [15]. Other times, hardware providers' Application Programmer Interfaces (APIs) can be used, recalling firmware counters to provide information in almost real time (e.g., Running Average Power Limit (RAPL) technology [19]).

NB-IoT implementations that are energy efficient are actively being studied as a way to increase overall energy efficiency in secured systems [2,4,7,10,11,12]. The primary goal of these studies has been to maximize the efficiency of the NB-IoT energy under a variety of conditions. All of these efforts, however, regard NB-IoT as a black box, and as far as we are aware, no study has been done in the literature that specifically addresses NB-IoT's underlying algorithms (e.g. hash) and tries to engineer them to use less energy.

2 Methodology

The SHA256 function of NB-IoT has been subjected to an Energy Complexity Model (ECM) [16]. First, we provide a quick explanation of how ECM is used with SHA256 to optimize energy. Next, we demonstrate the integration of this energy-optimized SHA256 in several phases of NB-IoT. Further details on the system and the experiments can be accessed here [9], or via the direct link to the project's GitHub page Hybrid Blockchain using ECM-SHA256 Algorithm.

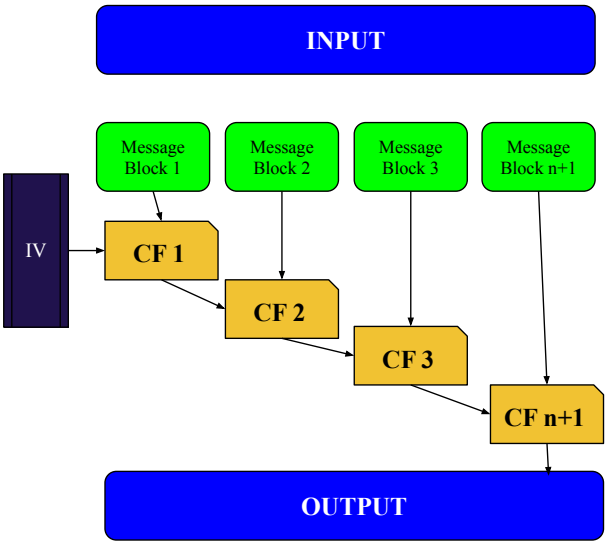

Fig. 1: SHA256 algorithm

2.1 Optimizing energy in SHA256

The ECM optimizes the energy consumption of any given algorithm $\mathcal{A}$ by parallelizing the data access of $\mathcal{A}$. ECM relies on (1) a memory model called double-data rate synchronous dynamic random access memory (DDR SDRAM) as its reference architecture [16], and (2) algorithm $\mathcal{A}$ being *block structured*, that is, $\mathcal{A}$' data accesses happen in blocks of fixed length. To satisfy (1), we used the DDR3 SDRAM as the memory model for our experiments, and SHA256 fits (2) well because it is inherently block-structured.

As illustrated in Figure 1, the SHA256 algorithm divides its input into fixed-size message blocks that are delivered sequentially to different compression methods. This block sequence is recognized according to the SHA256 algorithm's access pattern, which we engineer using the ECM.

The result of engineering the SHA256 algorithm based on ECM, Energy Optimized SHA256 (ECM-SHA256), is shown in Figure. 2. We used an 8-bank DDR3 SDRAM and use $I = 8$ as the parallelization index in our tests. This basically implies that we used the methods outlined in [17] to establish a virtual mapping that ensures that each size-8 access occurs across the eight banks for each set of eight consecutive block accesses in SHA256. Theorem 1 [3] proves that ECM-SHA256 performs not worse than SHA256 in terms of time.

Theorem 1. *ECM-SHA*256 *and SHA*256 *algorithms have the same computational complexity. [3]*

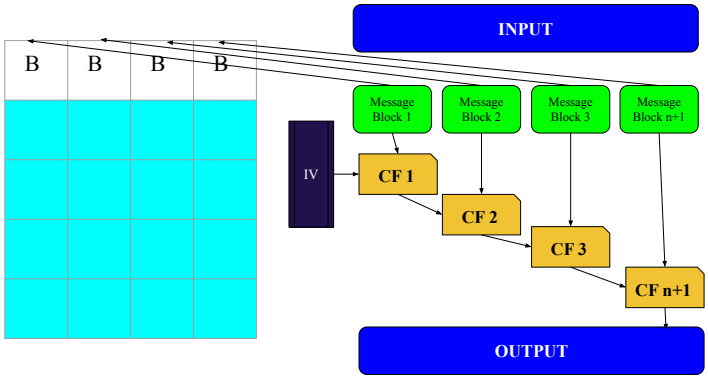

Fig. 2: Energy-optimized SHA256 algorithm (ECM-SHA256)

Furthermore, as long as the message blocks are processed in the same order as in Fig. 1, the output of SHA256 remains the same. This has been stated multiple times in the literature (e.g. [20]). This leads to Theorem 2.

Theorem 2. *ECM-SHA*256 *and SHA*256 *algorithms have the same integrity (security) level.*

2.2 Integrating ECM-SHA256 into NB-IoT

The objective of our experiment is to apply ECM-SHA256 to further optimize the energy consumption of NB-IoT devices in the proposed NB-IoT Blockchain Framework proposed by Mohan et al. [23]. With that motivation, ECM-SHA256 is integrated within the following three modules of the NB-IoT Blockchain Framework [23].

1. Unique ID Generation
2. Device Join Procedure
3. Device Transactions

1. Unique ID Generation When a device joins a cell, it is assigned a new Unique ID. *Unique ID* is used as an identifier in the NB-IoT Blockchain Framework. It is generated by the Data Server and sent to the NB-IoT client after authentication by the Authentication Server. The Unique ID Generation module consists of the NB-IoT Client, the Authentication Server and the Data Server, as described in Figure 3.
We integrate the ECM-SHA256 encryption algorithm by replacing SHA256 by ECM-SHA256 as shown in Figure. 3 (indicated by the blue hashes $h(\cdot)$).

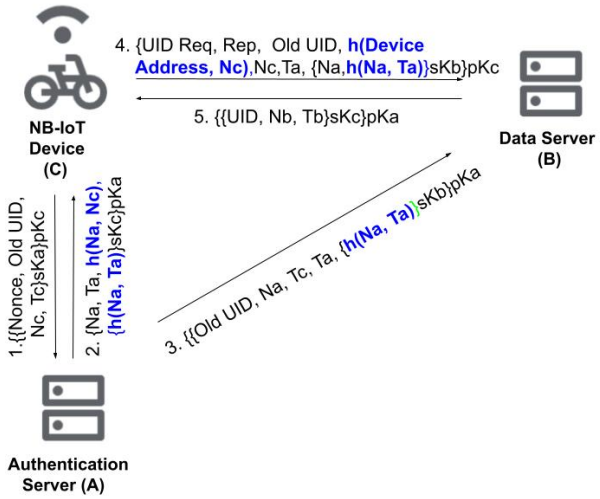

Fig. 3: Unique ID Generation

2. Device Join Procedure *Device Join Procedure* takes place when a device leaves one cell and joins a new cell. After Authentication from the Authentication Server, the NB-IoT device sends a *Leave Request* to the Home Base station along with the address of the Visiting base station. Home Base station approves the request and saves the smart contract along with transactions recorded in JSON format in Remix IDE. This file is then uploaded to the Inter Planetary File System (IPFS) and the resulting file hash is shared with Visiting Base Station. The NB-IoT device then sends a *Join Request* to the visiting base station. Similarly, Visiting Base station downloads the JSON file from Inter Planetary File System (IPFS), verifies the device using Unique ID and then sends 'New Unique ID' and 'Encryption Secret' to NB-IoT device. The Device Join Procedure module comprises the NB-IoT Client, Home Base Station, Visiting Base Station, and Authentication Server, as described in Figure 7.

We implement the ECM-SHA256 encryption algorithm again by replacing SHA256 with it, as indicated by the blue hashes $h(\cdot)$ in Figure. 7.

3. Device Transactions Only the base station in the NB-IoT Blockchain Framework is allowed to approve a transaction, which is then added to the blockchain. The device transaction process requires generating the SHA256 hash using the timestamp, nonce, unique id, reputation, message, and encryption secret. This hash is then sent to the base station by the NB-IoT device along with the hash parameters for approval. The Base Station will generate its SHA256 hash using the encrypted secret and reputation of the stored device together with the other transaction parameters received from the NB-IoT device. If

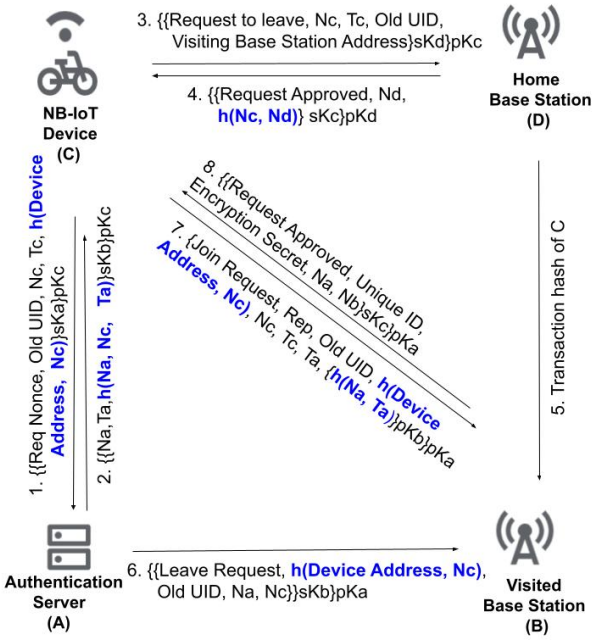

Fig. 4: Device Join Procedure

both match, the transaction is approved by the Base Station and added to the blockchain.

The Device Transaction module consists of only the NB-IoT Device and the Base Station, as described in Figure. 8. We again integrate ECM-SHA256 by replacing SHA256 by it, as shown in Figure. 8, indicated by the blue hashes $h(\cdot)$. Since messages of variable size can be passed during device transactions; we tested this module with message sizes of 64B, 128B, 256B, 512B, and 1024B.

3 Experiments

All of our experiments were conducted using two systems connected by wireless LAN: System I with IP address range `192.168.1.253/24` and System II with IP address range `192.168.1.71/24`. We furnish specification details of both systems. For more details on source code, runs, etc. see here [9], or the direct link to the project's GitHub page Hybrid Blockchain using ECM-SHA256 Algorithm.

3.1 System I

The system is equipped with a 64-bit quad-core Intel Core i5-8250U processor, running at 2900MHz and featuring a cache hierarchy of L1 256KB, L2 1024KB and L3 6MB. The system operates on Ubuntu Lubuntu 24.04 LTS (Noble Numbat), with 8GB of DDR3 RAM and 1TB of storage.

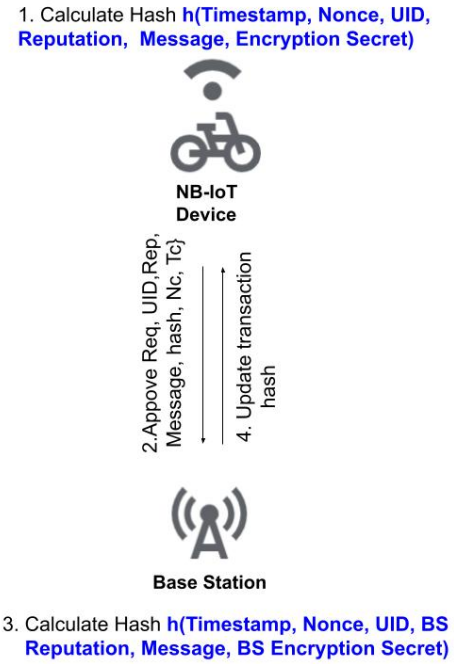

Fig. 5: Device Transactions

3.2 System II

The system is equipped with a 64-bit quad-core Intel Core i7-4600M processor, running at 2900MHz, and featuring a cache hierarchy of L1 128KB, L2 512KB and L3 4MB. The system operates on Ubuntu Lubuntu 24.04 LTS, with 16 GB of DDR3 RAM and 1TB of storage.

3.3 Network Protocol

Communication between the client and the server is ensured by using the User Datagram Protocol (UDP). In all our experiments, we set System I as the Client (IoT) and System II as Server (Authentication Server, Data Server and Base Station). Since NB-IoT devices are resource-constrained with limited bandwidth, we emulate this limitation by setting the transmission rate of a wireless interface on System I and System II to 0.144M using Python code.

3.4 Programming Languages

The ECM-SHA256 encryption algorithm is implemented in the C programming language, as it requires greater control over memory allocation in the operating system.

All modules of the NB-IoT Blockchain Framework are coded in Python. The SHA-256 encryption algorithms are written in C language due to the low-level memory management requirement, which is not possible in python. Thus, both the standard and ECM-SHA256 enryption algorithms are implemented in C language.
In order to call the SHA-256 encryption functions from Python NB-IoT Blockchain Framework modules, we utilize Python's CTYPES module. CTYPES allows us to call the C ECM-SHA256 functions from Python.

3.5 Energy Measurement Tool

To evaluate the energy consumption of the NB-IoT Blockchain Framework implementations, we employed pyRAPL, a software toolkit that measures the energy footprint of a host machine during Python code execution. PyRAPL leverages Intel's Running Average Power Limit (RAPL) technology to estimate CPU power consumption. Depending on the hardware and operating system setup, pyRAPL can measure energy usage in various CPU domains, including CPU (in μ Joules), and Duration (in μSeconds).
For performance evaluation, we first run the NB-IOT Blockchain Framework modules with the standard SHA-256 encryption algorithm. Similarly, we then run the NB-IOT Blockchain Framework modules with ECM-SHA256 encryption algorithm.
In order to take into account the noise from the pyRAPL energy measurement, we run the code for each module of the NB-IoT Blockchain Framework one thousand times for each module. PyRAPL calculates the mean value and exports the energy measurement results in csv format file.

3.6 Limitations

In our work, we have relied on the pyRAPL energy measurement toolkit to assess CPU usage and the duration of the involved modules. However, a limitation of pyRAPL is that it is only compatible with Intel processors. Our initial goal was to perform our experiments with the Raspberry Pi as a client in all the modules tested. This would have closely mimicked an IoT device with its inherent limitations in storage, RAM, and processing power. The Raspberry Pi does not host an Intel processor and therefore would not support the use of pyRAPL.

4 Results & Discussion

As previously stated, we have two different implementations of SHA256 in our experimental setup: the conventional implementation and the engineered one that uses ECM (which we refer to as ECM-SHA256 in this study). Tables 1 and 2 compare the energy consumption of NB-IoT modules using regular SHA256 (as done in [23]), and ECM-SHA256 (this work).

Table 1: Energy Consumption for Unique ID Generation and Device Joining procedures

Module	Client & Server	SHA256 CPU μJoules	SHA256 Time μSec	ECM-SHA256 CPU μJoules	ECM-SHA256 Time μSec	Savings CPU	Savings Time
Unique ID	IoT	774842	584787	750888	556404	3.1%	4.9%
	Authentication	1506117	585755	1436314	557470	4.6%	4.8%
	Data	1505859	585535	1436441	557417	4.6%	4.8%
Device Join	IoT	1316593	995397	1299641	980816	1.3%	1.5%
	Authentication	2442085	998613	2411952	983750	1.2%	1.5%
	Home	2438706	997650	2408627	982851	1.2%	1.5%
	Visiting	2435663	996787	2404560	981790	1.3%	1.5%

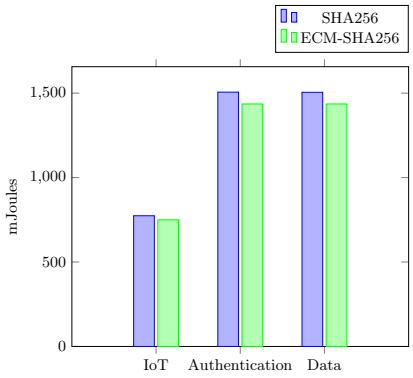

Fig. 6: Unique ID

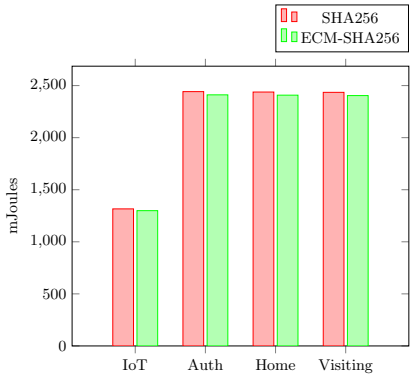

Fig. 7: Device Joining

Figures. 6, 7, 8, and 9 illustrate energy savings in milliJoules based on Tables 1 and 2. Again, we refer any interested reader for further experimental details to the GitHub link [9].

The first two modules of NB-IoT Blockchain Framework, Unique ID generation and Device join procedure, process fixed input. However, the device transaction processes messages of different sizes. Therefore, we tested its efficiency by passing messages of variable size i.e. 64, 128, 256, 512 and 1024 bytes. We observe the following improvements in the energy consumption of the NB-IoT system compared to [23].

We measured energy consumption of the CPU and the overall execution time for the three procedures (Unique ID Generation, Device Join Procedure, and Device Transactions) across Systems I and II.

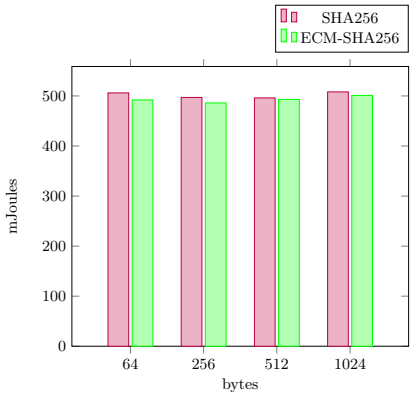

Fig. 8: IoT Device Transactions

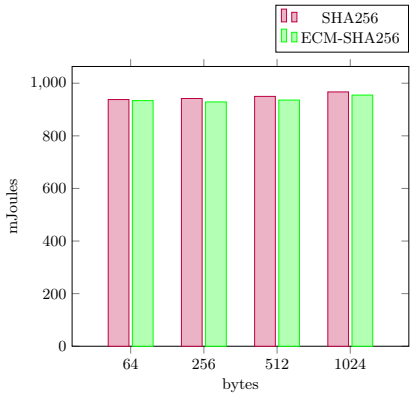

Fig. 9: Base Device Transactions

The first plot (Figure. 6) shows the savings in energy (milliJoules) and duration (milliSeconds) using ECM-SHA256 over SHA256 for the *Unique ID Generation* procedure. Table 1 indicates an improvement of up to 4.6% in CPU energy consumption and up to 4.9% in duration of the Unique ID Generation procedure across the IoT Client, Authentication Server and Data Server.

The second plot (Figure. 7) shows the savings in energy (milliJoules) and duration (milliSeconds) using ECM-SHA256 over SHA256 for the *Device Join* procedure. Table 1 indicates an improvement of up to 1.3% in CPU energy consumption and up to 1.5% in duration for the Device Join procedure across the IoT Client, Authentication Server, Home Base Station and Visiting Base Station.

It may be noted that the first two plots (Figure. 6 and 7) display the energy consumption of the IoT Client, which is approximately one-third of that of the Authentication Server and Data Server. This difference occurs because the IoT Client is running solely on System I, while both the Authentication Server and Data Server are operating on System II.

The third (Figure. 8) and fourth plots (Figure. 9) shows the savings in energy (milliJoules) and duration (milliSeconds) using ECM-SHA256 over SHA256 for the *Device Transaction* procedure. Table 2 indicates an average improvement of up to 2.8% in CPU energy consumption, and up to 2.2% improvement in duration for Device Transactions across the different message sizes (64, 256, 512, and 1024 bytes).

Given that NB-IoT devices are resource-constrained, with limited computational power, memory, storage, and bandwidth, we have restricted the message size in our test for IoT devices to a maximum of 1024 bytes. Within the given range, the results show less improvement in energy consumption as the message size increases.

Finally, taking into account the energy measurements from all three procedures (Unique ID generation, Device Join, and Device Transaction) of the NB-Iot Blockchain System, our results indicate an average of ($\approx 2.0\%$) improvement in

Table 2: Energy consumption of Device Transaction for NB-IoT

Module	Client & Server	SHA256 CPU μJoules	SHA256 Time μSec	ECM-SHA256 CPU μJoules	ECM-SHA256 Time μSec	Savings CPU	Savings Time
64 bytes	IoT	506382	416183	492220	414481	2.8%	0.4%
	Base	938172	416900	934001	414430	0.4%	0.6%
256 bytes	IoT	497560	417810	486206	412271	2.3%	1.3%
	Base	942181	418672	929539	412186	1.3%	1.5%
512 bytes	IoT	496230	414467	493296	412978	0.6%	0.4%
	Base	950389	415126	936617	413358	1.4%	0.4%
1024 bytes	IoT	508843	419403	501655	414143	1.4%	1.3%
	Base	967612	421241	955371	414166	1.3%	1.7%
Savings (Tables 1 and 2) in the NB-IoT Framework:						**2.1%**	**1.9%**

CPU energy measurement parameters and duration by accessing memory banks in parallel using the ECM-SHA256 hash algorithm in the NB-IoT Blockchain System (Table 2).

5 Conclusion

In this work, the overall energy consumption of the NB-IoT is reduced by engineering the underlying hashing algorithm (SHA256), which is based on the Energy Complexity Model (ECM) [16], and its integration into various NB-IoT procedures. By means of experimental energy measurements with different input sizes (e.g., Device Transaction process) of practical consequence, the ECM-enhanced implementation was compared with the standard implementation. According to the results, there can be energy savings of up to 2% in general and up to roughly 5% for a specific procedure (Unique ID Generation).
We utilized two different systems in our study. The IoT client was consistently hosted on System I, while the other servers—Data Server, Authentication Server, Home Base Station, and Visiting Base Station—were hosted on System II (see Section 3.3). It is worth noting that hosting all the machines on a single system would result in significantly higher energy savings due to reduced noise. These observations were made during an initial set of experiments that led to the development of this work.

References

1. Association, G., et al.: Nb-iot deployment guide to basic feature set requirements. GSM Association: London, UK (2019)
2. Bali, M.S., Gupta, K., Bali, K.K., Singh, P.K.: Towards energy efficient nb-iot: A survey on evaluating its suitability for smart applications. Materials Today: Proceedings **49**, 3227–3234 (2022)
3. Castellon, C.E., Roy, S., Kreidl, O.P., Dutta, A., Bölöni, L.: Towards an energy-efficient hash-based message authentication code (hmac). In: 2022 IEEE 13th International Green and Sustainable Computing Conference (IGSC). pp. 1–7. IEEE (2022)
4. Di Lecce, D., Grassi, A., Piro, G., Boggia, G.: Boosting energy efficiency of nb-iot cellular networks through cooperative relaying. In: 2018 IEEE 29th annual international symposium on personal, indoor and mobile radio communications (PIMRC). pp. 1–5. IEEE (2018)
5. Dorri, A., Steger, M., Kanhere, S.S., Jurdak, R.: Blockchain: A distributed solution to automotive security and privacy. IEEE communications magazine **55**(12), 119–125 (2017)
6. Flynn, K.: Narrowband iot. 3gpp. org, para **1** (2015)
7. Guo, Y., Xiang, M.: Multi-agent reinforcement learning based energy efficiency optimization in nb-iot networks. In: 2019 IEEE Globecom Workshops (GC Wkshps). pp. 1–6. IEEE (2019)
8. Harish, P.D.: Towards Designing Energy-Efficient Secure Hashes. Master's thesis, University of North Florida (2015)
9. Kakar, H., Roy, S.: NB-IoT Hbyrid Blockchain using ESHA-256 Algorithm (Sep 2024), `https://github.com/hafizkakar/ESHA-NB-IOT-V6`
10. Lee, J., Lee, J.: Prediction-based energy saving mechanism in 3gpp nb-iot networks. Sensors **17**(9), 2008 (2017)
11. Liang, J.M., Wu, K.R., Chen, J.J., Liu, P.Y., Tseng, Y.C.: Energy-efficient uplink resource units scheduling for ultra-reliable communications in nb-iot networks. Wireless communications and mobile computing **2018**(1), 4079017 (2018)
12. Migabo, E., Djouani, K., Kurien, A.: An energy-efficient and adaptive channel coding approach for narrowband internet of things (nb-iot) systems. Sensors **20**(12), 3465 (2020)
13. Mohan, V.S., Sankaran, S., Kumar, V., Achuthan, K.: Ep-cumac: Energy and performance-efficient integrity protection for narrow-band iot. Internet of Things **25**, 101004 (2024)
14. Pelaez, A.: Lorawan vs. nb-iot: A comparison between iot trend-setters. Ubidots. com, Feb **4** (2020)
15. Roma, C.A., Hasan, M.A.: Energy consumption analysis of XRP validator. In: Proc. of 2020 IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC). pp. 1–3. IEEE (2020)
16. Roy, S., Rudra, A., Verma, A.: An energy complexity model for algorithms. In: Proceedings of the 4th Conference on Innovations in Theoretical Computer Science. p. 283–304. ITCS '13, Association for Computing Machinery, New York, NY, USA (2013). `https://doi.org/10.1145/2422436.2422470`, `https://doi.org/10.1145/2422436.2422470`
17. Roy, S., Rudra, A., Verma, A.: Energy aware algorithmic engineering. In: Proc. 22nd IEEE Int. Symp. on Modelling, Analysis & Simulation of Computer and Telecommunication Systems. pp. 321–330 (2014)

18. Samara, G., Aljaidi, M., Al Daoud, E., Al-Safarini, M.Y., Alsayyed, G.M., Almatarneh, S.: Message broadcasting algorithm implementation using mobile long term evolution and narrow band internet of things over intelligent transportation system (its). In: 2022 International Arab Conference on Information Technology (ACIT). pp. 1–7. IEEE (2022)
19. Santos, M., Saraiva, J., Porkoláb, Z., Krupp, D.: Energy consumption measurement of C/C++ programs using Clang tooling. In: Proc. of 6th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications (2017)
20. Santos Jr, C.E., Silva, L.M.d., Torquato, M.F., Silva, S.N., Fernandes, M.A.: Sha-256 hardware proposal for iot devices in the blockchain context. Sensors **24**(12), 3908 (2024)
21. Sultania, A.K., Blondia, C., Famaey, J.: Optimizing the energy-latency tradeoff in nb-iot with psm and edrx. IEEE Internet of Things Journal **8**(15), 12436–12454 (2021)
22. Sunku Mohan, V., Sankaran, S., Buyya, R., Achuthan, K.: Leveraging fog computing for security-aware resource allocation in narrowband internet of things. Software: Practice and Experience (2024)
23. Sunku Mohan, V., Sankaran, S., Nanda, P., Achuthan, K.: Enabling secure lightweight mobile narrowband internet of things (nb-iot) applications using blockchain. Journal of Network and Computer Applications **219**, 103723 (2023). https://doi.org/https://doi.org/10.1016/j.jnca.2023.103723, https://www.sciencedirect.com/science/article/pii/S108480452300142X
24. Westin, J.: Evaluation of energy consumption in virtualization environments: proof of concept using containers (2017)