Low-Cost Tamper-Proof IoT Devices to Improve Data Origin Verification and Privacy in Blockchain-Based Energy Consumption Records

Daniele Orrù[1,†], Andrea Pinna[1,*,†] and Roberto Tonelli[1,†]

[1]*Department of Mathematics and Computer science, University of Cagliari, Via Ospedale 72, Cagliari 09124, Italy*

Abstract

Using sensor measurements and external data via the blockchain is the foundation of several investigated blockchain applications. These applications aim to take advantage of some of the key features of blockchain technology. However, cost, security, authenticity, and privacy problems may hinder the creation of real-world decentralized systems involving individuals, especially if public blockchains are utilized. In this study, we describe a simple secure and privacy-preserving architecture for registering energy consumption into blockchain logs using low-cost Internet of Things devices, based on ESP32. The devices are programmed with a custom version of the Web3 library and protected from cloning and tampering, as well as any attempts to obtain the private keys. The proposed system allows the device to forward signed transactions that guarantee the data provenance. Privacy protection is achieved by public-key cryptography of measurement data on blockchain, and guaranteeing that it has no connection with addresses or other data that could identify an individual but only the device. Finally, computational overhead, transaction and setup costs, and transaction throughput are estimated to evaluate a widespread application in real-world conditions.

1. Introduction

The utilization of blockchain technology for recording external data serves as the backbone for numerous utility applications[1] in various sectors, ranging from supply chain management[2] to healthcare[3] and smart city initiatives such as waste management[4], pollution monitoring[5]. These applications rely on the secure, immutable, and transparent nature of blockchain to ensure the integrity and authenticity of the data being recorded.

This is particularly relevant in the energy sector, where the consumption and production data of millions of users flows from smart meters to management systems[6, 7, 8]. However, the implementation of blockchain-based solutions in this scenario presents a set of challenges, originating from both the inherent characteristics of both blockchain technology and off-chain data sources[9]. Ensuring users' privacy remains paramount, necessitating robust privacy-preserving

mechanisms to safeguard sensitive information from unauthorized access or misuse. In a cyber-physical context, data sources are generally identified as IoT devices, capable of connecting to the internet and transmitting the data collected by sensors with a certain frequency[10]. In addition to privacy concerns, the transmission of data from IoT devices to the blockchain requires careful consideration of security measures. Various architectural approaches have been proposed to facilitate direct or indirect data transmission to the blockchain, each with its advantages and limitations [11, 12, 8]. Devices must be equipped with encryption capabilities to protect data integrity and confidentiality[13], mitigating the risk of tampering or unauthorized access. Furthermore, device security is a critical aspect that cannot be overlooked[13]. Implementing mechanisms to safeguard against tampering and unauthorized code modification is essential to maintain the integrity and reliability of the data being transmitted[14]. This involves implementing secure boot procedures, hardware-based encryption, and access control mechanisms to prevent unauthorized access to sensitive data[15, 16]. Recent works define secure and privacy-aware blockchain-based protocols for managing consumption and production data in Smart Grid, creating specific communication protocols [17, 18], and implementing specialized architectures [19, 20].

The novelty of this study lies in investigating the feasibility of simultaneously fulfilling the requirements of privacy, certified origin, and tamper resistance within a minimal public blockchain-based architecture that includes smart meters developed using low-cost IoT devices.[21].

The method of this study relies on using the built-in features of Ethereum-based smart contracts, such as creating logs via emitting events, verifying transaction signatures, and creating functions that can handle binary data. It also involves programming IoT devices to send transactions directly to blockchain nodes without any intermediary. Specifically, devices are programmed to use random-generated externally owned accounts, asymmetric encryption, flash encryption and secure boot. These methods safeguard the device against tampering and unauthorized access, enabling it to regularly collect and send signed consumption data while preserving individual privacy on public blockchains. The paper provides a detailed analysis of the system, including setup procedures, cost evaluation, and latency assessment, and discusses strategies for optimizing data throughput. While the presented study does not fully encapsulate the complexity of the problem, it serves as a robust proof-of-concept that preserve a simple architecture.

This rest of the paper is structured as follows: Section 2 describe the guiding requirements, the decision criteria and the implementation of the system; Section 3 reports the results in terms of transaction costs and performance, and the threats to validity; Section 4 draws conclusions.

2. Methodology

Our investigation revolves around a system wherein each user of an energy service is associated with an IoT device, generally known as a *smart meter*, which possesses the capability to transmit data regarding energy consumption at regular 15-minute intervals[22, 23], aligning with the time intervals observed in smart meters deployed by distribution service operators.

The study proceeded through several phases. Initially, we outlined the fundamental require-

ments and the system dynamics alongside delineating the features of programmable devices and the blockchain technology. Subsequently, the ensuing phase involved the programming both the device and the smart contract. Lastly, the culminating phase centered on the evaluation and assessment of the setup's performance.

2.1. Requirements and design

Two actors of the system are initially defined: the electric company (referred to as *company*) that is the entity capable of programming the devices to send transactions to a specific smart contract and accessing the data recorded on the blockchain; the user's device (referred to as *device*) that is the entity that regularly transmits cumulative consumption data.

The identified system requirements pertain to privacy, data origin verification, and tamper resistance. Specifically:

1. No one, including the company, can ever know the private key associated with a device.
2. The company is the only entity authorized to read the data recorded on the blockchain.
3. The device must be programmable to send signed transactions at a certain frequency.
4. The device must encrypt measurement data using a public key cryptography.
5. The device must be protected from code and data reading.
6. The device cannot be reprogrammed except by the company.

In addition, the device must have be low-cost (unit price of about 10 USD) and under no circumstances can the data on the blockchain contain information about the individual associated with the device.

The sequence diagram in Fig. 1 represents the main interactions and operations between the actors of the proposed system. The diagram illustrates the device setup phase and the main activity cycle, which repeats every 15 minutes. The setup phase involves flashing the device and generating the private key. The company will enable the device in the smart contract (and optionally provide the tokens necessary for its operation). The company can access the logs generated at any time. In the 15-minute cycle, the device will acquire and accumulate the value of the consumed energy, perform privacy-preserving encryption, sign the message, and send the transaction to activate a specific smart contract function that will emit an event and create a log, without the need to store values in the contract.

2.2. Device choice

Based on the requirements, several IoT devices were analyzed. Both programmable micro-controller units (MCUs) and commercial smart meters were included. Among the MCUs, the characteristics of ESP8266, ESP32 (original family), Raspberry Pi Pico W, and STM32WL were analyzed. These devices uses GPIO pins to gather inputs from the external environment and produce outputs. The comparison parameters are shown in table 1.

The Flash Encryption satisfies Requirement 5, because prevents physical reading of flash memory by encrypting data, and via the Secure Code Execution that protects against common attacks such as buffer overflow and code injection through memory segmentation. Secure Boot meets Requirement 6 by ensuring that only signed firmware can load, preventing unwanted
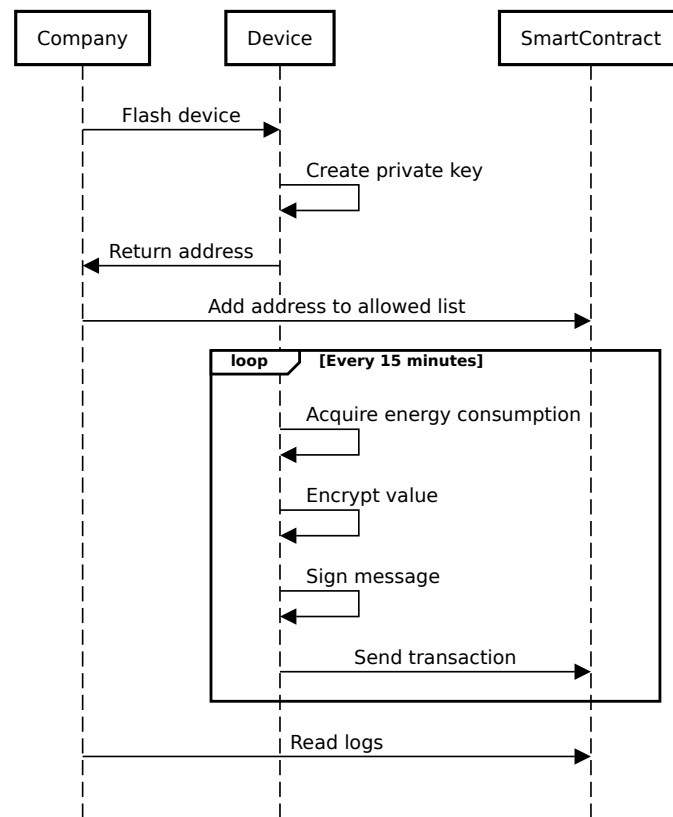
Figure 1: Sequence diagram of the operations within the system

modification. Furthermore, in previous studies, the ESP32 has demonstrated its potential to be used as a stand-alone meter [24, 25, 26]. For these reasons, the choice fell upon the ESP32[27].

2.3. System setup

The development and testing environment for this study comprise a local server hosting four nodes of a Hyperledger Besu blockchain.

The C++ code is utilized to program the device for blockchain connectivity and to execute all associated security operations. It employs the Web3E library for creating and sending signed raw transactions. The code is responsible for establishing connections and transmitting transactions

Table 1
Comparison of the features of examined MCUs

Feature	ESP8266	ESP32	Raspberry Pi Pico W	STM32WL
CPU Cores	Single core	Single or dual core	Dual core	Single or dual core
CPU Frequency	160 MHz	240 Mhz	133 MHz	48 MHz
GPIO Pins	17	34 to 38	40	43
Wireless Connectivity	Integrated WiFi	Integrated WiFi and Bluetooth	Integrated WiFi and Bluetooth	Sub-GHz radio (no WiFi/Bluetooth onboard)
Security Features	SSL/TLS, Lacks recent secure protocols	WPA3 support, SSL/TLS, Flash encryption, Secure Boot, Secure Protocols, Secure Code Execution, Cryptographic Accelerator	WPA3 limited support	AES hardware encryption, PCROP, public-key accelerator
Price(USD)	3 to 6	6 to 12	6,00	12 chip; 40 (complete dev. board)

to the blockchain and its smart contracts, as well as for executing the required cryptography operations such as hashing and signing. Notably, the Web3E library requires that the ESP32 exclusively connects to nodes using TLS/SSL. To comply with this security requirement, a standalone NGINX server with an SSL certificate and its corresponding key is installed on the machine hosting the node. This NGINX server efficiently directs ESP32 requests to the JSON RPC server of the Besu blockchain.

2.4. Implementation

Within the context of the use case, a single smart contract, coded in Solidity, using Remix IDE, has been deployed. The smart contract includes the following functions: *addPowerEntry*, which, upon receiving the encrypted amount of measured energy consumption, the hash, and the signature from the device account, verifies the origin of the message and emits an event (a blockchain log) containing this encrypted measurement and the device's address; *verifySigner*, which, given the hash of the encrypted value and the signature of that hash, returns *true* only if the hash's signature is valid and its signer corresponds with the public address of the sender;

The programming of the device was guided by the requirements and realized via Visual Studio Code IDE. To satisfy Requirement 1, the ESP32 device has been programmed in order to create a random Ethereum account in the first running. Its key is stored in the EEPROM of the device and secured from reading via Flash Encryption. At the first running, the device outputs the corresponding Ethereum address. The Company can use this address to allow the Device sending transaction to the smart contract. In this way, only the device itself knows this private key and uses it for sending signed transactions. As a consequence, the origin of the measurements is guaranteed.

To implement the second requirement, asymmetric encryption is used. The company include its public key in the device code, in order to allow the device to encode measurement data. The encryption phase makes use of the RSA module of the MbedTLS library[28], with keys of 2048 bit. To preserve privacy in possible on-chain operations, partial homomorfic encryption could

be used to sum encrypted consumption values directly on the smart contract. However, in our setup, we want to emphasize that it is not necessary to record the data in the contract storage, as the company can access the blockchain logs at any time and reconstruct the consumption of each user conveniently and practically.

The device program also includes a digital signature function via ECDSA that allows the device to transmit proof-of-origin. This could be useful even in cases where the network setup requires the use of network aggregators or other specific architectures to limit transaction overhead. Through ECDSA, the smart contract, using the *verifySigner* function, can verify the origin of the data and authorize forwarded requests. The system implementation is available on GitHub[1].

3. Results

In the test setup, the implemented system successfully executes the intended tasks. Specific experiments were conducted to evaluate performance in terms of costs and latency.

Regarding fixed costs, the use of low-cost devices for data transmission reduces costs by an order of magnitude compared to the use of commercial smart meters. For what concerns variable costs, it is possible to estimate the cost of use in public blockchains. For each transaction, each device, acting directly on the EVM blockchain, would consume approximately 43,000 gas units, and the transaction would weigh approximately 453 bytes including the sending of encrypted and signed data. Data operations include signature verification and event emission. The cost in the reference legal currency (USD) depends on the fee model of each blockchain and the corresponding native token's exchange rate. For example, considering a billing period of 30 days, 2880 transactions would be required from each device. The cost on Hedera, Avalanche, and the Ethereum Polygon sidechain would be approximately 9, 130, and 9 USD, respectively, for 30 days of activity on the network (calculated in March 2024). Note that using a subnet on the Avalanche ecosystem would reduce costs. Although it is expected that the company will maintain the device accounts, these costs would be borne by the users.

Regarding the device performance analysis, the time required for the device to create and sign a transaction was evaluated. Table 2 shows the typical times for executing each individual operation. It is noted that for each transmission, the device takes approximately 4 seconds. The most of time is due to the need to request the nonce from the network, the hashing operation, and the need to wait for transaction confirmation. These operations are by default handled by the Web3E library. The rest of the time is due to transaction creation, encryption, and signature production. It is observed that the device is capable of performing encryption operations in a relatively short time, demonstrating the effectiveness of the accelerator. Therefore, the device is suitable for transmitting data every 15 minutes. However, it would not be able to transmit data to the blockchain at a rate of one per second.

For the applicability of the solution in real-world applications, we consider a user base of approximately 40 million access points to the power grid distributed across about 2000 primary substations (data consistent with the Italian national territory, according to GSE[29] data). In the real system, for consumption accounting purposes, data is transmitted by meters every 15

---

[1]https://github.com/DenGames1211/SmartMeter/

Table 2

ESP32 execution time of single operations for creating, signing and sending transactions. The first operation include the request for the transaction count from the network (by using the library method EthGetTransactionCount).

Operation	Time (milliseconds)
Set Transaction Parameters (including nonce)	1560
Data Encryption (RSA)	46
Hashing (keccak256)	984
Signature	24
Send Transaction	1240

minutes. Assuming a random distribution of the data transmission moment, it is determined that approximately 45,000 transmissions occur per second. Taking this figure as a requirement, among the most widely used public blockchains, Solana (which, however, is not compatible with the system) would theoretically be capable of supporting such traffic. Hedera (EVM-compatible) could support an order of magnitude lower.

However, the use of an aggregator[12] for each area covered by the primary substation (also known as an *energy community*) would reduce the traffic by over 3 orders of magnitude, dropping to approximately 22 transactions per second, sufficiently low to be used on various EVM-compatible public blockchains such as Avalanche. As mentioned, in this case as well, the system would be protected from attacks aimed at compromising the certification of data origin through the signature verification mechanism. However, the robustness of the system would be compromised as there would be a single point of failure for each energy community, requiring investment in robustness. Alternatively, to reduce the transaction and rate it is possible to evaluate whether it is possible, in certain applications, to renounce sending data on a 15 minute basis and program the device to send cumulative measurements as a results of blocks of 24, 48 or 96 measurements (corresponding respectively at 6, 12 and 24 hours). It also leads to corresponding savings in transaction costs.

3.1. Discussion

The study, at this level of maturity, presents some relevant threats to validity. Foremost among them, although it was a desirable feature, the use of homomorphic encryption for summing up measured values in the smart contract was not experimented with. The adoption of existing solidity implementations such as fhEVM[30] would entail recalculating costs and execution times. Secondly, the analysis of applicability to public blockchains is based on data provided by documentation and third-party statistics, and the real impact that the application could have has not been evaluated. Thirdly, in this study the tests carried out were designed to verify the success of the experiment and evaluate the potential applicability. Future studies must include comprehensive validation of the results also using specific frameworks to test resistance to tampering and intrusion. In addition, the assumption of resistance to attacks aimed at identifying the device account's private key in this study is based on the countermeasures adopted by the manufacturer to address the issues identified in previous studies[31].

In addition, the experiments in this study utilize the original ESP32 family. Other versions such as the C3-series and the S2-series currently support safer and faster Secure Boot, Flash Encryption and other security protocols than the original ESP32. However, the real feasibility of their use in this application and their performance are deferred to future studies.

We also mentioned the use of private or permissioned networks to avoid transaction costs. From previous findings[32], it is noted that the use of the Hyperledger Besu permissioned blockchain as the supporting blockchain for this application would not be feasible with the number of transactions per second discussed earlier.

Finally, the novelties of this study builds upon concepts expressed and studied in previous works, as cited in the introduction. However, a comprehensive systematic comparison with existent solution will be necessary in a future extension of this work.

4. Conclusions

The study has demonstrated how a low-cost device, an ESP32, can be utilized to transmit privacy-aware encrypted energy consumption data of users through transactions with assured provenance, achieved through the random generation of the private key, and protection via flash encryption and Secure Code execution. The paper describes the requirements, device selection criteria, setup, and implementation of the system. Cost analysis has revealed a partial applicability of the system in some of the most important EVM-compatible public blockchains, both in terms of the high transaction rate for a national-scale territory and the transaction costs of approximately ten USD per month that would burden the citizen. Future studies envisage the completion of implementation through homomorphic encryption and experimentation.

Acknowledgments

References

[1] M. Krichen, M. Ammi, A. Mihoub, M. Almutiq, Blockchain for modern applications: A survey, Sensors 22 (2022). URL: https://www.mdpi.com/1424-8220/22/14/5274. doi:10.3390/s22145274.

[2] T. Van Nguyen, H. Cong Pham, M. Nhat Nguyen, L. Zhou, M. Akbari, Data-driven review of blockchain applications in supply chain management: key research themes and future directions, International Journal of Production Research 61 (2023) 8213–8235.

[3] A. Hasselgren, K. Kralevska, D. Gligoroski, S. A. Pedersen, A. Faxvaag, Blockchain in healthcare and health sciences—a scoping review, International Journal of Medical Informatics 134 (2020) 104040.

[4] P. Jiang, L. Zhang, S. You, Y. V. Fan, R. R. Tan, J. J. Klemeš, F. You, Blockchain technology applications in waste management: Overview, challenges and opportunities, Journal of Cleaner Production 421 (2023) 138466. URL: https://www.sciencedirect.com/science/article/pii/S0959652623026240. doi:https://doi.org/10.1016/j.jclepro.2023.138466.

[5] A. Kumar, B. Bhushan, S. Shristi, R. Chaganti, B. O. Soufiene, Blockchain-based decentralized management of iot devices for preserving data integrity, in: Blockchain Technology Solutions for the Security of IoT-Based Healthcare Systems, Elsevier, 2023, pp. 263–286.

[6] A. Chiarini, L. Compagnucci, Blockchain, data protection and p2p energy trading: a review on legal and economic challenges, Sustainability 14 (2022) 16305.

[7] M. Galici, M. Mureddu, E. Ghiani, G. Celli, F. Pilo, P. Porcu, B. Canetto, Energy blockchain for public energy communities, Applied Sciences 11 (2021) 3457.

[8] A. Ahmed, S. Abdullah, M. Bukhsh, I. Ahmad, Z. Mushtaq, An energy-efficient data aggregation mechanism for iot secured by blockchain, IEEE Access 10 (2022) 11404–11419.

[9] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, Blockchain technology in the energy sector: A systematic review of challenges and opportunities, Renewable and Sustainable Energy Reviews 100 (2019) 143–174. URL: https://www.sciencedirect.com/science/article/pii/S1364032118307184. doi:https://doi.org/10.1016/j.rser.2018.10.014.

[10] H. Rathore, A. Mohamed, M. Guizani, A survey of blockchain enabled cyber-physical systems, Sensors 20 (2020). URL: https://www.mdpi.com/1424-8220/20/1/282. doi:10.3390/s20010282.

[11] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, Y. Ma, Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities, IEEE Communications Magazine 56 (2018) 82–88.

[12] X. Luo, K. Xue, J. Xu, Q. Sun, Y. Zhang, Blockchain based secure data aggregation and distributed power dispatching for microgrids, IEEE Transactions on Smart Grid 12 (2021) 5268–5279.

[13] A. Dorri, S. S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for iot security and privacy: The case study of a smart home, in: 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2017, pp. 618–623. doi:10.1109/PERCOMW.2017.7917634.

[14] E. Gómez-Marín, L. Parrilla, J. L. Tejero López, D. P. Morales, E. Castillo, Toward sensor measurement reliability in blockchains, Sensors 23 (2023). URL: https://www.mdpi.com/

1424-8220/23/24/9659. doi:`10.3390/s23249659`.

[15] J. Lu, J. Shen, P. Vijayakumar, B. B. Gupta, Blockchain-based secure data storage protocol for sensors in the industrial internet of things, IEEE Transactions on Industrial Informatics 18 (2022) 5422–5431. doi:`10.1109/TII.2021.3112601`.

[16] A. A. Agarkar, M. Karyakarte, G. Chavhan, M. Patil, R. Talware, L. Kulkarni, Blockchain aware decentralized identity management and access control system, Measurement: Sensors 31 (2024) 101032. URL: https://www.sciencedirect.com/science/article/pii/S2665917424000084. doi:`https://doi.org/10.1016/j.measen.2024.101032`.

[17] N. Z. Aitzhan, D. Svetinovic, Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams, IEEE Transactions on Dependable and Secure Computing 15 (2018) 840–852. doi:`10.1109/TDSC.2016.2616861`.

[18] W. Wang, H. Huang, L. Zhang, C. Su, Secure and efficient mutual authentication protocol for smart grid under blockchain, Peer-to-Peer Networking and Applications 14 (2021) 2681–2693. URL: https://doi.org/10.1007/s12083-020-01020-2. doi:`10.1007/s12083-020-01020-2`.

[19] S. S. Hussain, S. M. Farooq, Blockchain based security and privacy scheme for smart meter communication, in: 2023 IEEE IAS Global Conference on Renewable Energy and Hydrogen Technologies (GlobConHT), 2023, pp. 1–6. doi:`10.1109/GlobConHT56829.2023.10087709`.

[20] C. Hu, Z. Liu, R. Li, P. Hu, T. Xiang, M. Han, Smart contract assisted privacy-preserving data aggregation and management scheme for smart grid, IEEE Transactions on Dependable and Secure Computing (2023) 1–17. doi:`10.1109/TDSC.2023.3300749`.

[21] F. Abate, M. Carratù, C. Liguori, V. Paciello, A low cost smart power meter for iot, Measurement 136 (2019) 59–66. URL: https://www.sciencedirect.com/science/article/pii/S0263224118312144. doi:`https://doi.org/10.1016/j.measurement.2018.12.069`.

[22] IBM, What Are Smart Meters? | IBM, https://www.ibm.com/topics/smart-meter, 2024.

[23] G. Dudek, A. Gawlak, M. Kornatka, J. Szkutnik, Analysis of smart meter data for electricity consumers, in: 2018 15th International Conference on the European Energy Market (EEM), 2018, pp. 1–5. doi:`10.1109/EEM.2018.8469896`.

[24] A. S. Salunkhe, Y. K. Kanse, S. S. Patil, Internet of things based smart energy meter with esp 32 real time data monitoring, in: 2022 International Conference on Electronics and Renewable Systems (ICEARS), IEEE, 2022, pp. 446–451.

[25] A. Othman, N. H. Zakaria, Energy meter based wireless monitoring system using blynk application via smartphone, in: 2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), IEEE, 2020, pp. 1–5.

[26] S. Gadekar, M. Pimple, S. Thopate, A. Nikam, Iot based smart energy meter using esp 32, in: Proceedings of the 3rd International Conference on Communication & Information Processing (ICCIP), 2021.

[27] Expressif, Esp32 Wi-Fi & Bluetooth SoC | Espressif Systems, 2024. URL: https://www.espressif.com/en/products/socs/esp32.

[28] Mbed-TLS, An open source, portable, easy to use, readable and flexible TLS library, and reference implementation of the PSA Cryptography API. Releases are on a varying cadence, typically around 3 - 6 months between releases., https://github.com/Mbed-TLS/mbedtls, 2024.

[29] GSE, Mappa delle cabine primarie, 2023. URL: https://www.rinnovabili.it/energia/politiche-energetiche/mappa-delle-cabine-primarie-gse-cer/.

[30] Zama, A Solidity library for interacting with an fhEVM blockchain., https://github.com/zama-ai/fhevm, 2024.

[31] K. M. Abdellatif, O. Hériveaux, A. Thillard, Unlimited results: Breaking firmware encryption of esp32-v3, Cryptology ePrint Archive (2023).

[32] L. Mostarda, A. Pinna, D. Sestili, R. Tonelli, Performance analysis of a besu permissioned blockchain, in: International Conference on Advanced Information Networking and Applications, Springer, 2023, pp. 279–291.