SEBS: A Secure Element and Blockchain Stratagem for Securing IoT

Varun Deshpande
LIGM, UPE, ESIEE Paris
Noisy-le-Grand, France
varun.deshpande@esiee.fr

Taniya Das
TCS Research & Innovation
Kolkata, India
das.taniya@tcs.com

Hakim Badis
LIGM, UPEM
Champs-sur-Marne, France
hakim.badis@u-pem.fr

Laurent George
LIGM, ESIEE Paris
Noisy-le-Grand, France
laurent.george@esiee.fr

***Abstract*—In the context of the Internet of Things (IoT), the issue of holistic data security is complex as data has to be secured at 3 critical points i.e., 1. Point of generation (IoT sensors), 2. Point of storage, 3. Point of usage (IoT actuators). Point 2 can be adequately addressed by Blockchain that offers remarkable immutability by storing data securely and transparently in a distributed setting, making it tamper-proof while in storage. However, blockchain alone cannot root out all the data security impediments in an IoT network as Points 1 and 3 still needs to be resolved. In this paper, we propose a novel Secure Element and Blockchain based stratagem called SEBS to effectively secure all 3 Points at once, thereby realizing light, holistic, efficient IoT data security in true sense. Further, we elaborate on SEBS implementation in a generic IoT network scenario and show how it improves the performance of critical security operations by as much as 31 times. Next, we address a niche problem by proposing a novel SEOVA or SE based offline verification algorithm to verify if the data received through intermediary really belongs to the blockchain.**

***Index Terms*—SEBS, SEOVA, IoT, Blockchain, Secure Element, Offline Verification.**

I. Introduction

Internet of Things or IoT, as a new technology paradigm, is not concretely defined. Several descriptions are found in various research works. However, some common notions can still be conceived from them. A thing in IoT can be: a person with a heart monitor implant, an automobile that has built-in sensors to alert the driver of possible malfunctions, or any natural or man-made object that can be assigned an IP address and can transfer data over a network. In generic terms, the interconnection between people and (physical) objects, and communication between them in a smart environment, without requiring human-human or human-computer interactions, can be broadly defined as Internet of Things [1].

IoT is proving as a revolutionary technology, with increasing applications in many diversified domains. Its applications range from automation in manufacturing, energy (e.g. Smart grids), healthcare management, urban life (e.g. Smart city), monitoring crop fields with sensors, automating irrigation systems, to tracking product flow in factories, remote monitoring of patients and managing medical devices.

The predictions by McKinsey & Company shows that the economic impact of IoT will reach up to USD 2.7 to 6.2 trillion by 2025 [2]. Such overwhelming numbers show that IoT is having a major impact on society. The worldwide acceptance of IoT does appear smooth but includes major flaws that need to be solved before its worldwide implementation. The major issue that IoT holds is of security and privacy. The vision of IoT is the formation of connected self-conscious devices. When almost everything is connected, this flaw will become more prominent. It is still an open issue that IoT is facing.

As more and more devices are connected, the average number of security vulnerabilities per device is alarmingly increasing. This has raised concerns not only among the research folks around the globe but also has awakened the governments of various countries regarding its fatal effects on the social sector, critical infrastructure, and national security, if damage and misuse of IoT generated data happens. A recent conference [3] in London by Interior, Homeland security and public safety ministry of Australia, Canada, New Zealand, UK and US in July 2019, have shown major concern about security challenges with regards to IoT. The conference ended with a collaboration signed among these nations to resolve this problem to protect their citizens.

This problem is not new and has been given importance since the early 70's [4], however, none have proved to give efficient full data protection to the IoT devices. The existing techniques such as secure protocols, access control, encryption, authentication, etc. suffer from lack of adoption. Well-known encryption protocols, such as RSA, prove to be very expensive when running on devices with limited computing capabilities [5]. Limited power and computational capability of small IoT devices are insufficient for processing secure protocols. Complexity and size of protocol have made security expensive [6]. Apart from protocol and algorithmic level implementations, some new add-ons such as Blockchain (BC) have been proved very helpful to store data giving it more immutability and traceability.

Several attempts have been made to use BC with IoT together, but the channel to send data to/from IoT sensors from/to the BC remains prone to security threats. Intruding with the data may cause severe results affecting people's privacy, the countries' economy and critical infrastructure. Therefore, to bridge the gap between the IoT sensors and BC, we propose to introduce a Hardware Security Module (HSM) like Secure Element (SE). Further, we present a completely novel Stratagem called SEBS (Secure Element Blockchain Stratagem) based on these technologies i.e., BC, SE for effective IoT data security.

We show how SEBS impressively uses both the technologies to create a secure, reliable, immutable environment for data exchange and storage. Next, we also propose an SE based Offline Verification Algorithm (SEOVA) to allow Remote IoT Devices (RID) to check if the received data belongs to BC without connecting to it. To the best of our knowledge, such an idea is not proposed anywhere in literature so far.

The paper is organized as follows: Sec. II and III deals with related work and concepts respectively. In Sec. IV, problem statement is described in detail while in Sec. V we explain how we solve it using SEBS. In Sec. VI, we elaborate on our novel algorithm SEOVA to certify BC data affiliation. Sec. VII presents performance and overhead analysis. Finally, in Sec. VIII, we conclude with a brief overview of future work.

II. Related Work

In the discourse of use of BC as an IoT security solution, notable works have been done in many diverse fields, utilizing the immutability, trustlessness features of BC along with automating the process using IoT. BC has been used in many different ways in different contexts.

In [7] authors have proposed an architecture for access control based on BC technology. A single Smart Contract (SC) is used to reduce communication overhead between the nodes along with providing access control information to IoT devices in real-time. While [8] employs a local and private BC for providing secure access control in smart homes. Further, a lightweight encryption technique is employed for achieving security in smart homes. In [9], authors point out the ability of BC to create, store, transfer data in a distributed, decentralized and tamper-proof way which holds a great practical value in the context of IoT security. It suggests the use of cloud or fog as a potential host for the deployment of BC as Service (BaaS) for IoT. Lastly, it concludes fog as a better platform.

In [10], the authors talk about security issues in IoT devices used for remote patient monitoring. To handle protected health information (PHI) generated by IoT devices, BC-based SCs are proposed for secure analysis and management of medical sensors. They have utilized Ethereum protocol to create a system where sensors write records of all events in the BC. This will support real-time monitoring and medical intervention while maintaining a secure record of those who have initiated the action. The use of IoT will also automate the delivery of notifications to all involved parties in a HIPAA compliant manner.

In the context of smart cities and smart vehicles, a framework called SpeedyChain is proposed in [11], utilizing the potential benefits of BC in trusted data sharing of sensors in an IoT environment. Here, a BC design is used to decouple data stored in the transactions from the block header, allowing the fast addition of data blocks. The rich data set produced by embedded sensors in vehicles is used to provide various services. BC is used for reliable Vehicle-to-Infrastructure communication while maintaining vehicle privacy by using a periodically changeable key. It has also included an access level for devices to manage the permission of vehicles.

Food safety is becoming a serious topic these days. In [12], the topic of smart agriculture is explored. They have used IoT devices to replace manual recording and verification, reducing human intervention effectively. Problems such as excessive use of chemical fertilizers and additives, heavy metal contamination, use of inferior raw material are addressed and solutions are given using BC and IoT. BC is used as a tamper-proof and trustworthy verification mechanism to store and access information about tracking and monitoring the whole lifespan of food production, including food raw material cultivation, processing, transporting, warehousing, selling, etc. involving large no. of untrustworthy business parties.

In the case of smart grids, IoT devices are enabling prosumers to trade energy directly without involving any centralized third party. However, threats such as forging energy transactions have risen in such cases. So a fully decentralized functionality is implemented using BC, where energy transactions are stored and energy auctions are carried out according to transparent rules implemented as SCs, hence visible to all those who are involved. BC ensures a guarantee against tampering [13]. BC and IoT have found remarkable use in the areas of Disaster management also. In [14], authors propose early detection of natural disasters through crowdsourcing based fog computing (CDMFC) model in IoT. Further, if a direct link to the fog layer is not available, the data is sent to the nearest smart IoT object forming a peer to peer (P2P) network realized through BC technology. Since BC has a remarkable immutability feature, the data is safe and unaltered which can be safely used to predict disasters.

In all the above works, a fundamental assumption is considered while using BC for securing data in an IoT environment is that the data coming from/going to IoT network is true, unaltered and of verified origin. But in real-life scenarios, this is not the case. Our work tries to fill this exact gap through the proposition of SEBS and SEOVA as an effective solution for holistic IoT data security.

III. Relevant Concepts

The core principles of SEBS and SEOVA are based on two technologies i.e. BC and SE. In this Sec., a brief idea is presented for both technologies followed by a comparison highlighting similarities and differences.

A. Blockchain

BC in simple terms can be defined as a chain of data blocks, linked to each other via cryptographic functions [15]. In most cases, the cryptographic function utilized is *hashing*. Each block in the BC contains the hash of its previous block thereby forming a strong chain. BC can also be defined as a form of Distributed Ledger Technology which achieves immutability through its block-chaining structure and storing the copy of ledger across each participating node in the BC P2P network. Having multiple stored copies of the ledger makes it reliable in achieving an immutable and a truly persistent storage.

The coherency across all the ledger copies is achieved through the concept of consensus. Consensus, in layman's term, is a collective decision-making process through which the participating nodes decide on changing the current state of the ledger (collectively). Some notable consensus algorithms are Proof of Work, Proof of Stake, Proof of Elapsed Time, Practical Byzantine Fault Tolerance, etc. In the most common structure of the BC, each block contains data in the form of transactions. Depending on the BC platform, the data in a transaction can be financial, IoT/sensors data, code (SCs). This versatility helps to apply the BC concept not only to secure financial transactions but also IoT data, computer code, etc. in a distributed setting.

B. Secure Elements

A SE is a tiny (in the order of $25mm^2$), programmable, tamper-resistant micro-controller that is manufactured using a "secure by design" approach. It supports various digest and cryptographic functions like SHA, RSA, ECC, AES, etc. It comes with a crypto-processor to speedup the execution of these functions. It is used as an HSM to provide Trusted Execution/Storage Environment (TEE/TSE). TEE enables tamper-proof execution of various security-critical cryptographic functions (e.g. encryption, key generation, signature, etc.) while TSE assures that the underlying keys (symmetric keys, private keys, session keys, etc.) used in these functions are safe and cannot be leaked [16].

A typical SE has limited memory ($<$ 1 MB ROM, $<$ 15 KB RAM) [17] and can store 3 to 5 small applications performing specific tasks. SEs are classified according to the Operating System they run i.e., Multos (Multi-OS) based and Java Card based. SEs have been used in various day-to-day utilities like in Chip-based bank cards (credit/debit), SIM Cards, Chip-based Government ID documents, ePassports, etc. They are also used in IoT domain for: guaranteeing data source, tamper-proofing records, verification/certification/identification, TLS tunnel establishment, sensor monitoring [18], etc. Because of the "secure by design" approach, SEs have an additional security checkpoint for programming to prevent unauthorized re-programming. As in the case of Multos SE, special programming certificates called Application Load Certificates (ALCs) are needed for programming it while in the case of Java Card SE, a key is required to reprogram it.

Commercial SEs are certified under international standards of Common Criteria - Evaluation Assurance Level (CC-EAL). Depending on the criticality of security applications, certified SEs from EAL1 (lowest) to EAL7 (highest) are available [16]. SE, when used as a standalone secure micro-controller, can secure data from sensors that are directly connected to it. To secure a bigger and complex system, SE is used as an HSM. In this case, cryptographic tokens within SEs can be created/accessed via Public-Key Cryptography Standards (PKCS) #11. PKCS #11 defines standards for APIs governing access to common cryptographic objects like X.509 Certificates, RSA/DES keys, etc.

C. Comparison

BC and SE, both as an applied concept, can be utilized effectively to secure the data. For BC, once the data is stored on it, the architecture makes it nearly impossible to change any data. Also, multiple ledger copies offer truly an immutable, persistent and secure storage of data. However, in the case of data-transactions, where the source is remote IoT sensors, there is no way to verify its authenticity. Thus, even if the data is secured, there is no guarantee of its authenticity. This makes the whole data securing process pointless. Further, given its high overhead, high latency, consensus requirement through the P2P network, its penetrability in any system is limited to upper levels only.

For SE, given its low overhead and ability to work offline in a standalone setting, it can penetrate to lower levels compared to BC where it can secure data (offer immutability) using cryptographic functions like encryption and signature, right at the sensor level. However, as it lacks a big storage space to offer persistence for immutable data, securing data remains incomplete. These two technologies, when combined intelligently, can mitigate the shortcomings of each other through their strong pros. To elaborate, SE's shortcoming of unavailability of persistent storage is mitigated by BC while BC's inability to verify the source of data transactions and the property of large overhead is mitigated by SE's lower level penetration and very low overhead. SEBS and SEOVA involve intelligently combining these two technologies to create a foolproof add-on system for effective IoT data security.

IV. Problem Statement

In the context of IoT, a reliable automatic decision with high accuracy can be taken only if the sensor data is secured, right from the point of its generation to the point of its storage and the subsequent point of its usage. This is a tricky task as IoT devices/sensors are constrained with limited processing power, memory, battery autonomy, storage. Any additional implementation of cryptographic functions for increasing security severely affects their performance. Further, in the IoT domain, the environment is completely heterogeneous making it extremely difficult to implement single security protocol across all remote entities. Nonetheless, the problem has been solved *partially* with the effective use of BC as secure and persistent storage for critical IoT data. However, the impediments need to be solved further for effective data security in IoT. For clarity, we segregate BC use-cases in IoT into two types of scenarios and explain their distinct impediments:

A. Scenario 1: Blockchain as Data Sink

In this scenario, BC is the last component in the data flowchart i.e. it acts as a data sink. Typically, for this scenario, data is collected through Wireless Sensor Networks deployed on the remote IoT site and subsequently stored on the BC for future use (decision/policy making, automatic payments, etc). A typical application of this scenario can be found in smart home use-cases that secure sensor data on BC. In this, even if the data is secured after storing on BC, the authenticity, tamper-proofness and origin of the data cannot be verified effectively as data is not secured at the generation point. The principle data aggregator from the remote IoT site aggregates all data from sub-aggregators and push it on the BC. As there are no measures to detect tampering, authenticity, etc. at sub-aggregator level, this data is effectively used as it is without any additional safeguards. These additional safeguards are important in some use-cases like in renewable energy smart-grids where payments directly depend on the output of the sensor values.

B. Scenario 2: Blockchain as Data Source

In this scenario, BC is the first component in data flowchart i.e. it is used as a data source. Typically, for this scenario, BC is used as a distributed secure immutable storage serving as a backbone database. The data is retrieved from this backbone database and certain actions are performed on the remote IoT site depending on the data value/state. BC-based access control systems are an apt example of this scenario. For this, even if the data is secured through its storage on BC, the end entity i.e. Remote IoT Device (RID) in this case depends on an intermediary for its retrieval and also lacks the resources needed to verify this data. This can have critical security ramifications in use-cases like BC-based firmware update systems where the end-point retrieves data from BC as a service and lacks resources to ascertain if the data it has is really coming from the BC. Further, there is also a need to develop an offline data verification technique for the same.

To summarize, the challenges from both scenarios need to be solved effectively to completely secure IoT data. This includes securing the data at 1. generation, 2. storage, 3. usage. However, BC, with its immutability feature, can only solve point 2 by providing secure and persistent storage BUT cannot secure the processes of adding/retrieving data to/from BC. For meaningful incorporation in IoT's ecosystem, BC's use would only make sense if point 1 and 3 are solved as well because there is no purpose in securing/using data whose origin and authenticity cannot be verified (RID $- >$ BC, BC $- >$ RID). To the best of our knowledge, all the previous related works solve any 1 of the 3 points for IoT data security. Further, it is an intricate task to solve all the 3 points at once given the limitation of RIDs which are severely resource-constrained.

In this paper, we propose a completely new stratagem called SEBS which tries to fill this exact niche. SEBS or SE and BC based Stratagem effectively solves all the 3 points at once for complete data securization thereby securing the whole chain of events in data security in the context of IoT. The building blocks of SEBS i.e., SE and BC are aptly put together to complement each other and dodge their respective shortcomings thereby creating a light, safe and secure channel to transfer IoT generated data to and from BC.

V. The SEBS

The main purpose of SEBS is to secure the data transactions at 3 points i.e. 1) point of generation, 2) point of storage and 3) point of usage. For this, SEBS leverages on SE and BC. Scenarios 1 and 2 (see Sec. IV) requires to secure point 1-2 and 2-3 respectively for meaningful data security in IoT. To avoid redundancy in explanation, we illustrate SEBS implementation w.r.t Scenario 1 and highlight the differences w.r.t. Scenario 2.

A. SEBS implementation in Scenario 1

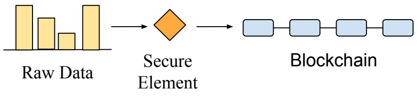

Fig. 1. The conceptual SEBS framework (data-flow).

The SEBS conceptual framework for Scenario 1 is presented in Fig. 1 which describes the data-flow. The concept behind it is that the data (raw) is passed through SE for tamper-proofing before it is stored on the BC. The implementational framework for the same is presented in Fig. 2. The process to use SEBS starts at the sensor level. The sensor can be either directly connected to the SE or through sub-aggregator, depending on the type of the sensor and required granularity in security. The raw data from the sensor(s) is sent to the SE. The SE then subsequently signs the data using the private key stored within it. This private key never leaves the SE and nor it can be copied (owing to "secure by design" architecture of SE).

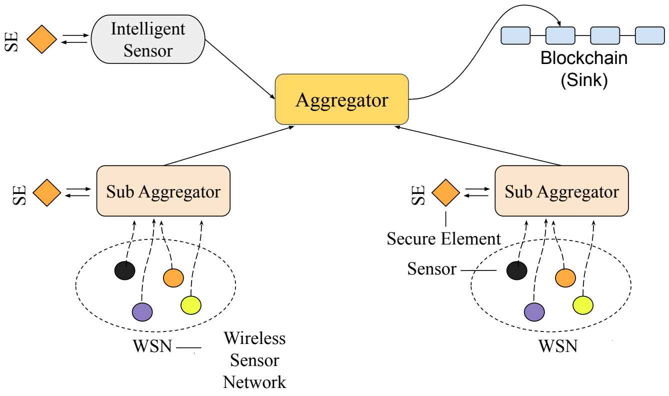

Fig. 2. SEBS implementation when BC used as a data sink (Scenario 1).

Once the endpoint receives this signed data and verifies the signature, it can be sure that indeed the data came from a particular sensor/sub-aggregator with whom this SE was associated. Further, if privacy is needed, the data can be encrypted with SE. Next, with SE, hardware integrity can be guaranteed as well, using a tamper detection circuit. If any tampering attempt is done, this circuit makes the tamper detection pin on SE high, following which the SE ceases all its functioning. Next, the data is then sent to BC (via sub-aggregator and/or aggregator) for secure and persistent storage. As the data is signed, it ensures the authenticity and integrity of the data while in transmission.

The end-point, which verifies this data during an audit, can, therefore, be assured about its stated place of origin, authenticity, and integrity during generation and transmission (point 1) and reading it from the BC verifies that no part of the data was deleted after storage (point 2). Thus, data obtained under SEBS cannot be denied/disputed by any of the concerned parties as whole process from data generation point to data storage point is secured. In audits, this helps to enforce the concept of *non-repudiable responsibility*. This applied concept is very useful in the context of IoT. The culpable party cannot deny the validity of data authenticity, integrity, origin, etc., on technical grounds during litigation, thereby resulting in faster dispute settlements.

B. SEBS implementation in Scenario 2

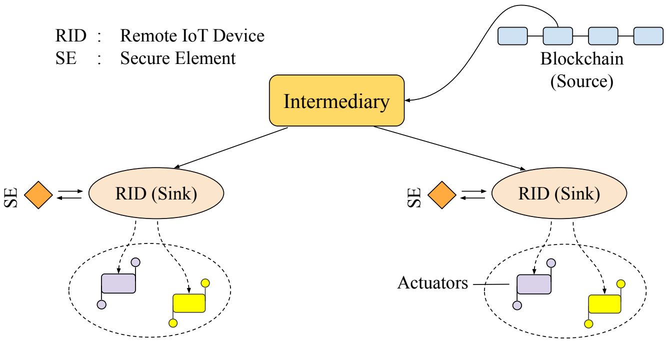

Fig. 3. SEBS implementation when BC used as a data source (Scenario 2).

The implementation of SEBS in Scenario 2 is very similar to Scenario 1 except that BC here is used as a data source while the RID is used as a data sink. The RID, in this case, the sub-aggregator (see Fig. 3), receives data from the BC through an aggregator. Given BC's immutability, the data is retrieved in its entirety (point 2). On reception, the data (signed) is sent to SE for signature verification and/or decryption. On success, the data origin and authenticity are thus ensured (point 3).

However, in most of the real-world implementations, RID cannot directly connect to the BC and depends on a trusted intermediary for information retrieval from the BC. This is frequent when BC is used as a service (BaaS). On reception, the RID has no means to verify if data came from BC. As data is added to the BC only after consensus (collective decision), any data retrieved from BC signifies that indeed, all participants had a consensus agreeing on that particular data. For example, in the use case of BC-based remote firmware update, a manufacturer's signed firmware stored on BC is more secured and trusted compared to another signed firmware from the same manufacturer as firmware on BC was added after consensus from the concerned parties (software developers, security watchdogs, etc.).

Hence, it is very crucial to not only ensure data origin (manufacturer in this case) but also data affiliation (i.e. it came from the BC). As RID cannot directly connect to BC to verify on its own, due to severe overhead constraints, an offline technique for verifying data affiliation is needed for realizing complete data security in IoT. In Sec. VI, we propose a completely new algorithm called SEOVA to achieve this.

VI. The SEOVA

SEOVA or SE based Offline Verification Algorithm is a novel algorithm to test BC data affiliation i.e. to determine if the data coming from BC through an intermediary (like in case of BaaS, RIDs), really belongs to the BC. In most BC platforms, the block validator, while creating a new block, signs it with its private key. This signed block is then disseminated across the BC network and added to the BC after consensus.

In a typical BC data verification process, to verify if a particular data comes from the BC, the block validator's signature in the block containing the particular data are verified. The verification is successful if the signature is verified using the public key from the list of registered validators. However, this approach has several flaws:

- Need to maintain a list of public keys of all the validators in the BC network.
- Need to update this list after every new block formation.
- Need to connect to the BC to execute the above 2 actions.
- Offline verification hence not possible.
- Critical overhead ramifications for RIDs that are resource-constrained and connected to BC via an intermediary.
- Requires trust in the intermediary.

To solve these drawbacks, within SEOVA, we propose to alter this commonly used single signature process in favor of a novel double signature block formation process in conjunction with the apt use of SE to implement the same. A generic physical architecture for implementing SEOVA is given in Fig. 4. Each validator (node) in the BC network has a dedicated SE [19].

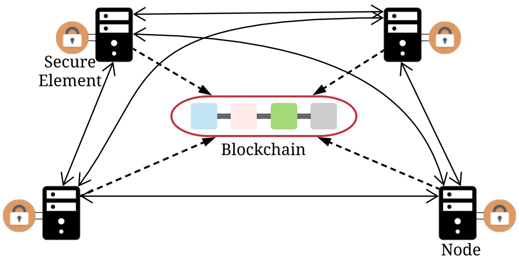

Fig. 4. Generic physical architecture for SEOVA.

Each of the dedicated SE is instantiated by a trusted entity using a secure process elaborated in [19]. During the instantiation, 2 private keys (validator key P_V, blockchain key P_B) and supporting codes are securely injected into the SE. P_V is the unique private key for the particular validator and P_B is the shared private key common to all validators of the BC. The corresponding public key (P_B^{pub}) of P_B is injected into the SE of the concerned RIDs (see Fig 3).

The process for SEOVA starts at the validator's end. When new data D is to be added to the BC, the validator creates a new block B containing this data. This block is then signed by its dedicated SE twice (i.e. with P_V and P_B). This double signed block B_S^2 is then disseminated across all the participating nodes, verified and finally added to the BC after consensus. Algorithm 1 illustrates the process in detail which is followed by each node (validator) whenever they want to add new data to BC.

Referring to the Scenario 2 (Fig. 3), whenever, RID receives data (encapsulated in a block) from the BC via a trusted/non-trusted intermediary, it simply verifies the signature using P_B^{pub} key to ascertain data's BC affiliation. This approach has a wide range of advantages:

- No need for a trusted intermediary.
- No need to connect to BC for verification.
- No overhead on RID as calculation-intensive tasks delegated to SE.

Algorithm 1 Block formation for SEOVA implementation.

```
Input: new data $D$ for insertion in BC;
Output: double signed block $B_S^2$ containing new data;
  Initialization
  get the new data $D$;
  verify data origin $v \leftarrow ECDSA\_verify(D)$;
  if $v == failed$ then
      $echo$(“Data origin cannot be verified”); $exit$;
  end if
  Block Formation
  create new block $B$ with $D$;
  create new block header $H_B$ for $B$;
  insert previous block’s $HASH(H_{B-1})$ in $H_B$;
  insert $merkel\_tree$, $timestamp$ in $H_B$;
  Double Signing Using SE
  sign $S_V \leftarrow ECDSA\_sign(H_B)$ with $P_V$;
  sign $S_B \leftarrow ECDSA\_sign(H_B)$ with $P_B$;
  Block Finalization
  insert $S_V, S_B$ in $H_B$;
  $B_S^2 \leftarrow append(H_B, B)$;
  return $B_S^2$;
```

- Validator cannot leak or copy P_B given SE’s secure by design architecture.
- No need to change key when validators maintaining BC change their individual P_V/new validators are added/removed.
- SE can be programmed to detect hardware tampering attempts and will subsequently cease to function if tampered.
- Identity theft of validators through leaked P_V-P_B, is effectively prevented as keys inside SE cannot be replicated.

VII. Overhead Analysis

In SEBS and SEOVA, the novelty lies in the additional intelligent usage of SE with traditional blockchain-based data security approaches. Since SE is installed as an additional component at the RID and the BC validator level, we perform overhead analysis at both levels.

A. Computational Overhead

RIDs, being severely constrained, are either incapable or acutely disadvantaged to perform resource-intensive cryptographic operations securely. With SE as an HSM add-on, they gain this ability at a cost of increased computational overhead for maintaining communication with SE (serial/I2C/SPI). This little (increased) overhead cost is still significantly lower against the native implementation of cryptographic functions. Further, at the BC validator level, the delegation of resource-intensive processes like encryption, decryption, signature and its verification significantly reduces the computational overhead on the validator node.

B. Timing Overhead

To measure the timing overhead, we set up a testbed consisting of RID, BC node, and a SE. We used Arduino Nano 3.0 and Raspberry Pi 2 Model B as RIDs. For the BC node, we used Dell XPS with an Intel i7-8550U processor. For SE, we used Multos M5-P19 [20]. To quantize the timing overhead, we performed 100 iterations of ECDSA signature and verification on each device. The averaged results with standard deviation are presented next.

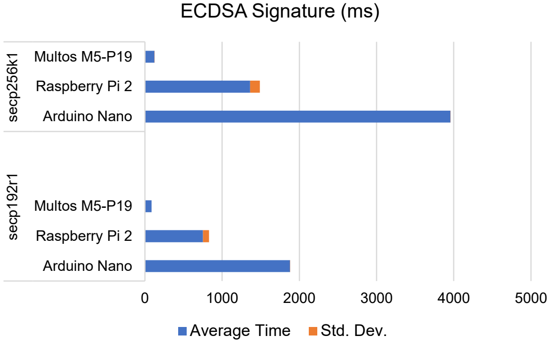

Fig. 5. ECDSA signature time cycle, RIDs vs SE.

Given its secure architecture and specialized cryptoprocessor, SE performed much better compared to RIDs. The performance improved further for bigger ECC curves. For ECDSA signature (Fig. 5), Multos SE was up to $31x$ and $10x$ faster compared to Arduino Nano and Raspberry Pi respectively. Similarly, for verification (Fig. 6), it took up to $25x$ and $16x$ less time respectively (secp256k1).

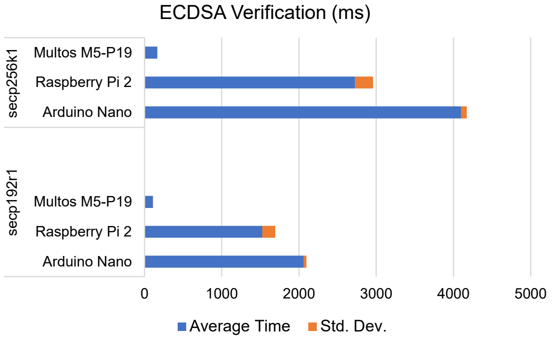

Fig. 6. ECDSA verification time cycle, RIDs vs SE.

When compared to BC node validator, SE was slower by about 70% for signature (Fig. 7) and 60% for verification (Fig. 8. However, SE with its TEE and TSE was deterministic with negligible standard deviation. The increased time overhead, even though significant in relative terms, accounts only for a few hundred *ms* and gives unmatched security advantages offered by the SE.

When combined, the equivalent timing overhead of the SEBS and SEOVA is reduced as the timing overhead at the RID level is significantly reduced in magnitude compared to its little increase at the BC node level.

C. Memory Overhead

Assuming we use ECC [21] with a 256-bit curve, at RID level, one private key of 32 bytes for signature and one public key of 64 bytes for verifying data’s BC affiliation is needed. As these keys are stored in SE (TSE), the memory overhead is reduced by 96 bytes. Similarly, for BC validator, one public key of 64 bytes for verifying data source and 2 private keys

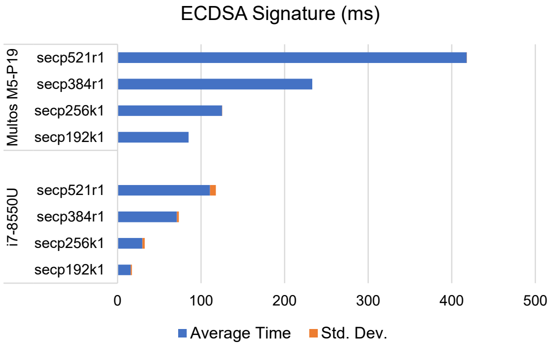

Fig. 7. ECDSA signature time cycle, BC node vs SE.

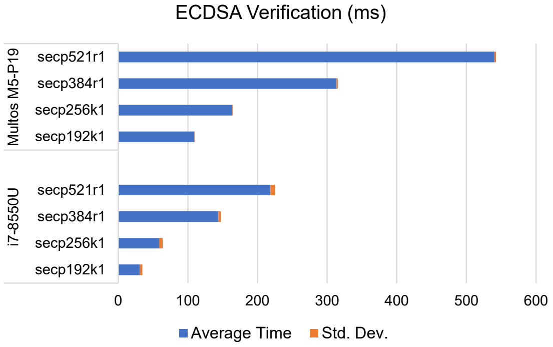

Fig. 8. ECDSA verification time cycle, BC node vs SE.

of 32 bytes each, are needed for double signature. Therefore the memory overhead is reduced by 128 bytes. Although with a small factor, the SE reduces overall memory overhead.

D. Cost Overhead

Since additional hardware component installation is required at the RID level and the BC node validator level, the cost overhead is increased. The cost of SE varies with the certification level. The cheapest available SE (non-certified) costs 0.5€ while EAL7 certified SE costs 10€. Since, in our proposition of SEBS and SEOVA, one SE is needed at each of the 3 critical points (see sec. V), the cost overhead is between 1.5€ to 30€ depending on the required certification level.

In a nutshell, our proposed SEBS and SEOVA reduce computational, timing and memory overheads greatly in a given IoT system. Further, when gauged against the benefits, they clearly outweighs the little increased cost overhead.

VIII. Conclusion and Future Work

In this paper, we have proposed a Secure Element and Blockchain based stratagem called SEBS which is capable of effectively eliminating all the recognized data security issues, in the context of IoT. A framework is also explained for the implementation of SEBS. Finally, an innovative lightweight algorithm called SEOVA is also proposed to verify if the data belongs to the BC when received through an unsecured/untrusted channel, at RID. Performance evaluation showed that the SEBS-SEOVA proposition can increase the performance of critical security operations by as much as 31 times, all while reducing computational and memory overheads. Future work will be to comprehensively expand this work for its implementation with smart contracts and other distributed ledger technologies.

References

[1] A. Riahi, Y. Challal, E. Natalizio, Z. Chtourou, and A. Bouabdallah, "A systemic approach for iot security," in *2013 IEEE international conference on distributed computing in sensor systems*. IEEE, 2013, pp. 351–355.

[2] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs, *Disruptive technologies: Advances that will transform life, business, and the global economy*. McKinsey Global Institute San Francisco, CA, 2013, vol. 180.

[3] The Home Secretary, UK, "Statement of intent regarding the security of the internet of things," 29-31 July 2019.

[4] E. Bertino and E. Ferrari, "Big data security and privacy," in *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*. Springer, 2018, pp. 425–439.

[5] A. Singla, A. Mudgerikar, I. Papapanagiotou, and A. A. Yavuz, "Haa: Hardware-accelerated authentication for internet of things in mission critical vehicular networks," in *MILCOM 2015-2015 IEEE Military Communications Conference*. IEEE, 2015, pp. 1298–1304.

[6] S. Babar, A. Stango, N. Prasad, J. Sen, and R. Prasad, "Proposed embedded security framework for internet of things (iot)," in *Wireless VITAE 2011*. IEEE, 2011, pp. 1–5.

[7] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[8] A. Dorri, S. S. Kanhere, R. Jurdak, and 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)*. IEEE, 2017, pp. 618–623.

[9] M. Samaniego and R. Deters, "Blockchain as a service for iot," in *2016 IEEE International Conference on Internet of Things (iThings)*. IEEE, 2016, pp. 433–436.

[10] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of medical systems*, vol. 42, no. 7, p. 130, 2018.

[11] R. A. Michelin, A. Dorri, M. Steger, R. C. Lunardi, S. S. Kanhere, R. Jurdak, and A. F. Zorzo, "Speedychain: A framework for decoupling data from blockchain for smart cities," in *MobiQuitous 2018*. ACM, 2018, pp. 145–154.

[12] J. Lin, Z. Shen, A. Zhang, and Y. Chai, "Blockchain and iot based food traceability for smart agriculture," in *Proceedings of the 3rd International Conference on Crowd Science and Engineering*. ACM, 2018, p. 3.

[13] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, and V. Sassone, "A blockchain-based infrastructure for reliable and cost-effective iot-aided smart grids," 2018.

[14] A. Rauniyar, P. Engelstad, B. Feng *et al.*, "Crowdsourcing-based disaster management using fog computing in internet of things paradigm," in *2016 IEEE 2nd international conference on collaboration and internet computing (CIC)*. IEEE, 2016, pp. 490–494.

[15] V. Deshpande, H. Badis, and L. George, "Btcmap: Mapping bitcoin peer-to-peer network topology," in *2018 IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, Sep. 2018, pp. 1–6.

[16] V. Deshpande, L. George, and H. Badis, "Safe: A blockchain and secure element based framework for safeguarding smart vehicles," in *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*, Sep. 2019, pp. 181–188.

[17] P. Urien, "Racs: Remote apdu call secure creating trust for the internet," in *Collaboration Technologies and Systems (CTS), 2015 International Conference on*. IEEE, 2015, pp. 351–357.

[18] V. Deshpande, L. George, and H. Badis, "Pulsec: Secure element based framework for sensors anomaly detection in industry 4.0," in *9th IFAC Conference Manufacturing Modelling, Management and Control MIM 2019*. Elsevier, 2019.

[19] MAOSCO Limited, "Guide to loading and deleting," *MAO-DOC-TEC-008 v2.28, Page 5-22*, 2017.

[20] Multos, "Ml5-p19 and mc5-p19 on infineon sle78 platform." [Online]. Available: https://www.multos.com/products/approved_platforms/MIR/multos_international/m5_p19

[21] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," *International journal of information security*, vol. 1, no. 1, pp. 36–63, 2001.