Blockchain-enabled secure crowdsensing for trackside infrastructure information collection and validation in railway signalling data preparation

Ganesan Muniandi

Abstract
Rail operators around the world are adopting advanced control systems for railway signalling and train protection to improve their safety performance. In these re-signalling projects, the information about the already installed signals, point machines, track circuits etc. along the trackside infrastructure is essential for the configuration of advanced signalling equipment. The conventional information collection method which is based on unmanned aerial vehicles, helicopters etc. has the access limitations in terrains such as tunnels, hilly regions, river bridges, dense forests etc. To overcome these challenges, this paper proposes the railway vehicular crowdsensing method in which one group of trains participates in this information collection and another group of trains confirms the validity of the collected information. For information exchange between the trains and centralized server, this study chooses the permissioned blockchain-based information transaction mechanism to ensure the trustworthiness. Furthermore, the railway markup language-based blockchain databases are included for information immutability, crowd information integration, and the automatic execution of data preparation signalling rules. Finally, the case studies are carried out to analyse the adequacy of the proposed combination of crowdsensing based secure trackside infrastructure information collection and validation, permissioned blockchain-enabled information transaction and blockchain databases in the proposed signalling data preparation process.

1 INTRODUCTION

With the growing demand for environment-friendly, easily accessible, affordable, safe, and long-distance mass mobility solutions, the high-speed and semi-high-speed rail networks are expanding across the world in recent years. Conventionally, the safe driving of these trains is guaranteed either by the loco pilot or by the national automatic train protection system. To improve the operational interoperability, system performance, safety etc., several railway operators have been progressively adopting advanced signalling solutions such as the European Railway Traffic Management System (ERTMS) [1], Positive Train Control (PTC) [2] system etc.

ERTMS [1] trackside equipment such as radio block centre, euro balise, lineside electronic unit etc. and PTC trackside equipment [2] such as a wayside interface unit, transponder, switch monitoring systems etc. include both the generic application software and the specific configuration data [2]. Therefore, for the proper working of these kinds of equipment, the specific data is as critical as the generic software. For each signalling equipment, this configuration data is prepared based on the trackside infrastructure information [3].

This infrastructure information [3] comprises track topology, track elements, railway operation, and control system elements etc. The basic information about each track i.e. the beginning and end of the track, connection for siding track and switches etc. are defined in track topology [4]. The examples of track elements [4] include speed changes, track gradient changes, track radius changes, tunnels, bridges, level crossings, platform edges etc. The railway operation and control system elements [4] can contain information about signals, train circuits, balises etc.

This infrastructure information can be directly mapped into railway Markup Language (railML) [4]. The railML is an open, Extensible Markup Language (XML) based data exchange format for data interoperability of railway applications [5]. The big...
data modelling of railway infrastructure using railML is demonstrated in the projects such as 'Intelligent integration of railway systems' (InteGRail) and 'Automated and cost-effective railway infrastructure maintenance' (ACEM-Rail) [6].

Similarly, in the case of interlocking [7] configuration data preparation, the specific signalling rules are applied over the railML information. Overall, the railML has been proven to solve the problem of signalling data modelling [7, 6]. In the context of proposed data preparation process, this study also chooses the railML database as the base. Section 2 introduces the background of the conventional data preparation process.

2 BACKGROUND

The steps involved in typical railway signalling data preparation process [3] are (a) trackside infrastructure information collection, (b) trackside signalling data design which includes application of signalling principles over collected information in the form of railML, (c) data validation, (d) data verification. For this data preparation process, there exist, in general, two approaches [3], namely, single-chain and double chain processes. The single-chain process mandates the formal method approach [8]. There are two possible variants in the double-chain process [3]. For variant-1, the two chains i.e. chain-1 and chain-2 use the same specification to generate the same signalling data. Finally, the generated data outputs are compared using another tool. In the variant-2, the chain-1 generates the output data and chain-2 regenerates the input information from chain-1 output data. In the end, input and regenerated input files are compared using another tool. This paper chooses a double chain variant-1 process due to its simplicity to realize the double chain process with the common specifications.

As shown in Figure 1, [3], the variant-1 double chain process is realized in four layers as (a) the information collection layer, (b) optional intermediate data verification layer, (c) data design and validation layer, and (d) data verification layer. Arrows in Figure 1 represent sequential information flow from one layer to another layer from bottom to top. Several step numbers are indicated inside the ‘yellow’ circle.

At first (Step 1 of Figure 1), the information about the trackside signalling infrastructure is collected either manually or automatically. For manual information collection, the data design chain-1 and chain-2 teams (Figure 1) refer to the existing track and signalling plan drawings that are provided by the railway operators. The automatic information collection is carried out using helicopters [9], Figure 1, or Unmanned Aerial Vehicles (UAVs) [10], drones [11], Figure 1, or airborne laser scanning [12]. The major drawback of these airborne vehicles is their limited access across the terrains such as tunnels, hilly regions, river bridges, dense forests etc. Hence, the railway survey system like GPSinfradat uses Global Navigation Satellite System (GNSS) systems along with inertial, radar, and high-speed cameras [13]. This collected information is stored in the form of railML along with camera data [13]. Similarly, Fugro’s train-mounted survey system [11] measures absolute track position and collects laser point cloud information of the entire rail corridor for generation of topographical survey information.

Secondly (Step 2 of Figure 1), the optional intermediate information verification rules layer checks the correctness of the collected information instantaneously. This check can be done either manually by applying information verification rules or automatically by using tools [8]. Once the information from both chains is consistent with each other, it is declared as valid for further usage. Else, the information collection layer will be alerted for the possible re-collection.
Thirdly (Step 3 of Figure 1), the data design is a key step in which the dedicated ERTMS or PTC signalling principles/rules are applied over the collected information. There is also a data validation in this layer to perform the tests on the designed data.

Finally (Step 4 of Figure 1), the formal data verification is completed using the dedicated tools, for example, [8]. This step helps to ensure whether the designed data satisfies the properties of the system under deployment for a specific project.

Though the existing data preparation method caters to the current needs, still there is the scope of improvements. The main drawbacks are: (a) majorly manual execution, (b) sequential execution (bottom-up), (c) each step is distinct, (d) mostly offline, (e) time-inefficient as the defect is detected only in later stages which may escalate the cost (f) lack of provision for concurrent information integration even when the signalling upgrade is executed by multiple contractors over different railway lines across the country.

With the major drawbacks on one hand and the recent technological evolutions, on the other hand, this paper proposes a novel railway signalling data preparation method. The purpose and significance of adopting specific technologies into the conventional process are detailed in Section 3.

3 | PURPOSE AND SIGNIFICANCE

To overcome the drawbacks exist in the conventional data preparation process, this paper proposes a novel data preparation process that uses the combination of real-time (a) decentralized information collection, (b) secured information transaction, (c) centralized information processing and integration into railway, (d) immutable databases, (e) decentralized guided information validation, (f) finally, the automatic application of conventional data design, validation, and verification rules, for the first time.

In the context of decentralized real-time information collection (a) and guided validation (d), the crowdsensing [14–17] can be the real enabling technology. In general, crowdsensing [14] refers to a technique where a large group of individuals with sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest.

The crowdsensing can be broadly classified into two categories, namely, mobile crowdsensing [14] and vehicular crowdsensing [15] based on how the information is collected.

The current generation of smartphones evolved from the conventional communication device to edge computing machines in the context of mobile crowdsensing of environmental, infrastructure, and social application data [14]. Due to the high and cheap availability of smart mobile phones, this mobile crowdsensing has almost negligible deployment costs [18]. Even though the applications of mobile crowdsensing are wider from healthcare, social, e-governance, education, environmental monitoring, to transportation, the challenges are also huge [19]. Addressing these challenges, several frameworks, architectures, and design strategies were proposed [20, 21] for mobile crowdsensing. In another category, the vehicular crowdsensing was demonstrated in road vehicles for the non-safety critical applications such as parking navigation, road surface monitoring, traffic collision reconstruction etc. [15]. Compared to mobile crowdsensing, the railway vehicular crowdsensing may be the better candidate for the information collection by considering the limited access or awareness of railway infrastructure to typical mobile phone users.

The future of rail data collection [11] at Network Rail involves the complex technologies such as inertial navigation systems, satellite positioning, photogrammetry, mobile data connections and the latest high-speed laser scanning systems. This automation of data collection not only provides added value and also can have a big impact on the railway surveying [11]. At the same time, the addressing the major problems associated with geospatial engineering’s fast-changing software applications, data storage and data security are also most needed [11]. Network Rail collects detailed information about its track and therefore the surrounding features, like bridges and tunnels [22]. The information is then analysed to assess clearances between trains and therefore the infrastructure around them, which is vital to safety.

Though major advances made in these crowdsensing methods, still their adaptation in railway safety-critical application data preparation is impracticable in its current form due to crowd trustworthiness, and information trustworthiness, transaction security, integrity and storage security issues etc. [23–25, 11]. However, these typical problems can be addressed through the use of blockchain technology [23, 26]. By definition [23], ‘blockchain is the mechanism that allows transactions to be verified by a group of unreliable actors. It provides a distributed, immutable, transparent, secure, and auditable ledger’.

By considering the possible improvements in the traditional data preparation process and the concerns in crowdsensing, this paper proposes a novel railway vehicular crowdsensing that incorporates the blockchain-based transaction (b) for decentralized real-time information collection and validation.

Depending on the access rights, the blockchains can be broadly categorised into public or permissionless blockchain, private or permissioned blockchain and hybrid type [27]. The hybrid blockchain is a combination of the public and private blockchain. In the case of permissionless blockchain, any participant can read, write and participate in transactions and in the consensus without the need for special authentication or authorisation. The examples of permissionless blockchains are Bitcoin and Ethereum [27]. The major drawback is that it is more prone to attacks such that the anonymous participants can have several identities to influence the consensus.

In the case of permissioned blockchains, the authentication is mandatory for any participant to gain access and to use the system resources, such examples include Multichain, Hyperledger Fabric, Parity, BigChainDB, InterPlanetary, Corda and Quorum [27]. Permissioned blockchains have permissioned blockchains the dedicated hardware module to manage the identity, privacy, confidentiality and auditability within the system [27]. In the context of this study, permissioned blockchain is suitable [28] due to a) the participation of the trains in the crowdsensing is based on invite only i.e. the registered trains only can participate in the proposed crowdsensing method. Thus, each train will be identified using
the unique pair of private and public keys [28] in the permissioned blockchain. b) As permissioned blockchains need their own nodes to validate transactions, the proposed work involves the choice of few crowd i.e. few trains to participate in the validation of information transactions.

For information transaction, the permissioned blockchain relies heavily cryptography as similar to Public Key infrastructure (PKI) in terms of the usage of private/public key pairs [28]. In comparison, PKI focuses only on how the key is managed and data is signed than about how the data is stored and shared, which can also be performed by permissioned blockchain. In ERTMS [29], Key Management Centre (KMC) and PKI are used for the key management and data signature creation. The KMC can play the role of a Certification Authority (CA) in PKI [30]. KMC generates the valid certificates that will guarantee the identity of the ERTMS entities. However, the option of CA as KMC is not available during information collection stage prior to ERTMS or PTC deployments. Hence, it is required to incorporate the self-generation of keys than using of CAs. Unlike PKI, the major advantage of permissioned blockchain is that it does not need a KMC like CAs to issue certificates [31]. Hence, these equivalent certificates are self-generated by the crowd themselves compared to PKI and the trust about the collected data is arrived from a network of validating crowd in this study.

In the case of centralized information integration (c), the information immutability has to be satisfied, i.e. once the train stores the information into a railML database that should not be altered by other trains. The adoption of blockchain technology into the railML centralized database also can be an answer to this requirement. Hence, the blockchain hash is included in railML [32, 33] database to lock the stored information from modification by the trains.

In summary, five major contributions of this paper in the novel data preparation process are:

1. Railway vehicular crowdsensing based decentralized trackside information collection to enable the multi train information collection which can emulate Fugro’s train-mounted survey system [11] if it is installed in several trains for simultaneous information collection and validation.
2. Permissioned blockchain-enabled information transactions between the train and trackside centralized server and vice versa to ensure the data security requirements [11]
3. Blockchain-integrated immutable railML databases at a centralized server for enabling data storage with security [11]
4. Information fusion of railML blockchain raw data using nearest neighbour and probabilistic data association algorithms for enabling the fusion of collected multi-train information.
5. Railway vehicular crowdsensing based decentralized tracksid information validation so as to be compliant with the transaction validation of permissioned blockchain such as Multichain [34].

The rest of this paper is organized as follows: Section 4 describes the architecture of the vehicular crowdsensing enabled, blockchain-enabled trackside railway signalling data preparation process. The various components used in the proposed architecture are also detailed in this section. The case studies along with simulation results are discussed in Section 5. Finally, Section 6 concludes the summary of the study. In the rest of the sections, the term ‘crowdsensing’ is used in place of a ‘railway vehicular crowdsensing’.

4 ARCHITECTURE OF BLOCKCHAIN-ENABLED RAILWAY SIGNALING DATA PREPARATION

As seen in Figure 2, the proposed signalling data preparation process comprises of following layers:

- Crowd layer
- Crowdsensing information collection layer
- Permissioned blockchain layer
- Crowdsensing information validation layer
- Data design layer
- Data validation layer
- Data verification layer

Arrows in Figure 2(a) represent information flow from one layer to another layer from bottom to the top except for the data validation layer. The various components of the architecture are discussed in the remaining subsections of this section as per Figure 2(b). Several step numbers involved in the proposed process are indicated inside the ‘yellow’, ‘blue’ and ‘green’ circles in Figures 2 and 3. The yellow circles indicate the steps involved in ‘information collection’ and the ‘blue’ circles show the ‘information validation’ steps. The remaining steps are drawn in ‘green’ circles.

4.1 Crowd layer

Crowd layer consists of two groups of trains, i.e. one group of trains acts as the crowd for the trackside infrastructure information collection and another group of trains performs the information validation to realize the permissioned blockchain. This crowd can be a loco-pilot or co-pilot of the train. Furthermore, the train [35] or railway vehicle trolley [36] with automatic cameras and trackside infrastructure object recognition capability [37] can also be a crowd. These rail vehicles will be equipped with the odometer, accelerometer, gyroscope along with GNSS etc. to perform crowdsensing functionalities. The crowd layer is marked in the Steps 1, 8 in Figures 2 and 3.

4.2 Crowdsensing information collection layer with passive crowdsensing

As mentioned in Figures 2 and 3(a), in this layer, the train (1 to n) collects trackside infrastructure information (Step 2) and sends them to the trackside centralized server by participating in passive crowdsensing. In the case of passive
crowdsensing [38], ‘the crowd shares the information on their own to extract and integrate relevant information regarding a specific task’. As the crowd involved is continually required for signalling information collection and validation, the participatory sensing mode [39] is required. Participatory crowdsensing [14] requires ‘the active involvement of individual train to contribute to the reporting of trackside infrastructure information related to a data preparation process’.

In this passive participatory crowdsensing, the crowd collects the information, for example, as mentioned in Table 1 [40, 38]. The registered loco pilot or co-pilot can visualize and identify the trackside infrastructure information via the driver machine interface (DMI) to collect or validate the specific information. In another possibility, DMI of the train is programmed to take pictures using dedicated cameras and confirm the trackside infrastructure objects using advanced algorithms [41-43].
The neural network [41] or deep learning [42, 43] based algorithms at edge device can be used for the classification of specific railway trackside infrastructure objects such as signal etc.

The DMI, sensors, and local processing unit formulate an “edge device” as represented in Figure 2(b). In general, these edge devices are responsible for ‘information gathering, processing, and filtering of sensor data, as well as data aggregation’ [44].

4.3 | Permissioned blockchain layer with elliptic curve digital signature algorithm

Once the information is collected, it is sent to the centralized server over the permissioned blockchain layer (Step 3 of Figures 2, 3(a)). For this blockchain layer, the Elliptic Curve Digital Signature Algorithm (ECDSA) [45] is chosen which is commonly adopted in permissioned blockchains such as R3 corda [46] and Multichain [34] in this paper. In general, ECDSA
### TABLE 1  Typical trackside infrastructure information

| Trackside infrastructure object name | Sensors | Information |
|---------------------------------------|---------|-------------|
| Gradients—gradient changes            | Gyroscope/crowd manual input | pos, absPos, slope |
| Curvature—radius changes              | Gyroscope/crowd manual input | pos, absPos, radius |
| Permanent speed restriction—speed changes | Crowd manual input | pos, absPos, vMax |
| Level crossing—level crossings        | (Side) Camera/crowd manual input | pos, absPos, |
| Inter track—track                    | Gyroscope/crowd manual input | trackBegin pos, absPos,trackEnd pos, absPos, |
| Switches/point machines—switch, connection | Gyroscope/crowd manual input | length, pos, absPos, switch connection with course, maxSpeed, orientation |
| Signals—signals                      | (Side) Camera/crowd manual input | pos, absPos, dir, type |
| Platforms—platform edge               | (Side) Camera/crowd manual input | pos, absPos, length, side |
| AFTEC track circuit/axle counter—train detection elements | (Side) Camera/crowd manual input | pos, absPos |
| Euro balise—balise                    | (Bottom) Camera/crowd manual input | pos, absPos, dir |

***absPos value is calculated either using the combination of GNSS and Odometer/accelerometer automatically.***

is a cryptographic algorithm used by permissioned blockchain such as Multichain [34] to ensure that funds can only be spent by their rightful owners. It uses cryptographic elliptic curves (EC) to generate key-pair of Private key (PrK) and Shared key (ShK) [47].

Each train (Crowd) generates its PrK which is a secret number. This key is maintained as confidential by the train. The PrK is the 32 bytes or 256 bits unsigned integer which is generated using any “random number generator” within the range of 1 to \(2^{256}\). The ShK is a number that is calculated from PrK, i.e. \(\text{ShK} = \text{ECDSA}(\text{PrK})\), but vice versa is not possible, i.e. \(\text{PrK} \neq \text{ECDSA}(\text{ShK})\). It has a size of 64 bytes or 512 bits.

The collected information is signed using PrK before it is sent to the centralized server. At first, the double-hash of the information is created by calculating the SHA-256 (Secure Hash Algorithm-256) of SHA-256 of the information, i.e.

\[
\text{Hash of information} = \text{SHA-256 (SHA-256 (information))}
\]

Then, the hash of the information is signed using the PrK of the crowd, i.e.

\[
\text{Signature} = \text{Sign (Hash of information, PrK)}
\]

Once the information is received, the centralized server uses an ShK to verify whether the signature is valid or not, without knowing PrK (Step 4). The verification is done in two-sub steps. In the first substep, the hash of the information is recreated as,

\[
\text{Recreated hash} = \text{SHA-256 (SHA–256 (information))}
\]

The second substep involves verifying the signature using ShK of the corresponding crowd,

\[
\text{Verification} = \text{Verify (Recreated Hash of information, Signature, ShK)}
\]

If the hash and the regenerated hash of the received information are the same, then it is declared as correct. With the help of signature and verification using ECDSA, security, and integrity issues are resolved. This way only the registered crowd can participate in crowdsensing.

Similarly, when the centralized server allocates the information validation task to the individual crowd, it signs the information with its private key. The registered individual crowd, which has the shared key of the centralized server only can participate in this validation task.

#### 4.4  Data design layer with blockchain databases

The major activities performed in this layer (Step 4 of Figures 2, 3(a)) are (a) to push the collected information into railML blockchain raw database (b) to evaluate the raw data for the outlier detection and retain only the non-deviant information, (c) to use one of the fusion algorithms, namely, single nearest neighbourhood (NN), k-nearest neighbourhood (kNN) or probabilistic data fusion (PDA) to combine the non-deviant information and to upload it into railML blockchain database, (d) finally, to apply the ERTMS or PTC specific signalling principles/rules automatically to transform the railML blockchain information into specific configuration data.

A typical structure of the railML blockchain database is described in Appendix-A.1. The railML blockchain raw database is the subset of the railML blockchain database which stores the collected information from the crowd. As seen in Figure 3(a), the railML blockchain database stores the fused information. As the entire collected information is located on a centralized server in the form of a railML blockchain database, it is easier to apply signalling rules and also perform the data validation and verification.

#### 4.5  Crowdsensing information validation layer with active crowdsensing

As shown in Steps 5 to 11 of Figures 2, 3(b), the information validation is executed as the ‘guided validation’, i.e. active
crowdsensing. This step is compliant with the transaction validation of permissioned blockchain such as Multichain [34]. In the case of active crowdsensing [38], ’the crowd generates the information based on requests to actively design and collect data for a specific task and then integrate the information’ (Step 5).

In this paper, a centralized server guides the crowd (train n+1 to 2n of Figure 3(b)) with the information to be validated, i.e., it transmits the upcoming trackside infrastructure information, for example, as per list in Table 1 over the blockchain layer (Step 6). For each information validation task, the centralized server signs the information to be verified with its private key. The registered individual validation crowd uses a shared key of a centralized server to verify whether it is intended to it (Step 7). Crowd confirms the trackside information (Step 8) using the provided button interface in DMI application manually or automatically [41–43].

Finally, the steps involved in Section 4.3 section are again repeated by the crowd (Steps 9, 10) to send the “validation” status message, i.e. Pass (1) or Fail (0). The centralized server updates the rawML blockchain database with the validation status as Pass or Fail (Step 11). When the number of valid information collection reaches the designed limit (design choice), the validation is confirmed as completed and the corresponding flag is set in the database. This concludes the transformation of the collected information into reliable data. Then, the further steps namely, data validation and data verification can be executed directly or the specific signalling rules can be applied before data validation and data verification steps.

5.1 | Case-1: Permissioned blockchain-enabled information transaction and processing to update rawML blockchain raw database

The objective of this case study is to numerically demonstrate the substeps involved in permissioned blockchain enabled information transactions, information processing at the centralized server, and finally, to store the properly received information into the railML blockchain raw database as described in Figure 3(a). The following assumptions are made in this case study:

(a) Each train which is part of identified crowd collects the trackside infrastructure information.
(b) Each train is fitted with the cameras at the top, side, and bottom of the first vehicle for imaging the trackside infrastructure as per Table 1. (c) Each train is equipped with DMI as an edge device (d) for each information collection, the convergence is reached by comparing the actual camera image with the reference image [41–43]. (e) The image identification, classification, and confirmation are not detailed in this study as they were well studied in the works of literature, for example, refs. [41–43].

Each train sends this collected information in the form of a blockchain transaction by signing using its PrK (Step 3). For this action, each train uses ECDSA along with cryptographic elliptic curves (EC) to generate the key-pair of PrK and ShK, as similar to permissioned blockchain, for example, Multichain [47]. For example, the train-1 generates the random confidential private key (256 bits, 64 hexadecimal digits) as,

$$\text{Prk} = \text{8B8AD}F52488D\text{D660F83637C8E}A14A\text{D}F574B382F43AE10\text{C4104CE8BC1B75CE7}$$

The shared key (512 bits, 128 hexadecimal digits + ‘04’ prefix) of the train 1, which is equivalent of PrK is given as, 

$$\text{ShK} = \text{04F61092B1080185B4EA652A89116AECC}$$

The key is transmitted in the encoded form as, 

$$\text{Signature} = \text{ECDSA Sign}(\text{SHA-256(\text{SHAK-256 (Information)})), \text{PrK})$$

The SHA-256 of the train-1 information is formulated as, 

$$\text{SHA-256(Data)} = \text{D65F300A396A672FE11FF9310DAE0B904B666C9007A4780C18EA9790D305387C9}$$

The signature of the train-1 is computed as, 

$$\text{Signature} = \text{3044022017A40B128D6D53F01733E0777E0A}7A2F3AC800E2C3460DA7F42B5F2A07F84022032488C}1FC6603D6517594827E04D0EC3B462C188F2B098C8570C49D439A7C9$$

The signature is transmitted in the encoded form as, 

$$\text{Encoded Signature with Base64} = \text{MEQICIBeqC6KNDVpWf2Pg}834K6pL2e4Ac4ZGD90sBYw+EI4A0KHvsm.A9ZRsZCgTQ7DC0Yi48rCyFmsdQ58kg==$$

Similarly, the encoded form of ShK is sent to centralized server as, 

$$\text{Encoded ShK with Base64} = \text{MFywEAYHKoZIqg0CAQYFC4EEAAdQqAE9bC8AGFOPkumG+awzKja397}$$

**4.6 | Data validation and verification rules**

These layers are the same as the conventional data preparation process. The data validation layer is involved with the automated testing of reliable data with generic and project-specific rules. The data verification layer further verifies whether the final data adheres to the expected signalling principles for the specific project configuration. This layer is not detailed further as they are almost the same as the conventional signalling data preparation process except the automatic execution of rules.

5 | CASE STUDIES WITH RESULTS AND DISCUSSIONS
At the centralized server, the proper information reception (Step 4) is confirmed in two substeps. First, the hash of the information is recreated using SHA-256. Then, the ECDSA verification is performed using ShK as,

\[
\text{Signature Verification} = \text{ECDSA Verify (SHA-256(SHA-256((Information)), Decoded Signature, Decoded ShK)} = \text{true}
\]

to confirm the blockchain transaction as the valid one. If the information is invalid or cannot be verified with any crowd’s public key, then, the collected information is rejected, and further, the analysis is done to check whether any malicious information attack was attempted.

Once the signature verification is successful, the train identification is confirmed as ‘train-01’ by a centralized server. Then, the information is split using ‘|’ separator, and the Infrastructure type is identified as ‘Signal’. Then, the infrastructure id and track id are extracted. Finally, as shown in Figure 4, the centralized server updates the railML blockchain raw database with the data,

\[
\begin{align*}
\text{signal hash} & = \text{“d01f7878b487fe6402e37582af7c419298ea3dd3c”} \\
\text{train id} & = \text{“train_01”} \\
\text{time} & = \text{“2020-04-19:09-05”} \\
\text{pos} & = \text{“10.21”} \\
\text{dir} & = \text{“up”}
\end{align*}
\]

under the railML element,

\[
\text{railML/infrastructure/id} = \text{“i_01”}/\text{tracks/track/id} = \text{“tr_01”}/\text{ocsElements}
\]

The hash is calculated using the “SHA-1” algorithm. The actual time of reception of the collected information is also added while writing it into the railML blockchain raw database. Similarly, each train updates its information as provided in Figure 4. Once the information is integrated into this raw database, then, the next step is information fusion.

### 5.2 Case-2: Information fusion

The objective of this case study is to demonstrate numerically the substeps involved in the integration of information from a group of trains (Step 4). The information fusion is carried out into two substeps. The first substep involves the computation of information centroid. In general, the centroid is the arithmetic mean position of all the information samples. Secondly, the fusion algorithm is applied over railML blockchain raw data to reject the outlier information using computed centroid and to combine only the non-deviant information.
The following assumptions are made in this case study: (a) One group of 10 trains (crowd) participates in the information collection. (b) Each train is equipped with onboard sensors such as GNSS, gyroscope, accelerometer etc. as mentioned in [48]. (c) As per [48], for each train information, $3\sigma$ location accuracy of 1.5 m is achievable.

### 5.2.1 Information centroid computation

The centroid calculation is initiated only when the number of collected information reaches the designed limit. For example, with 100 collected information per train, the total number of measurements, $N = 1000$ has to be reached. Consider the signal at the absolute position of 10 m which is marked with the legend ‘Expected Centroid’ in Figure 5 and Table 2. To inject the location uncertainty ($3\sigma$) around signal location information, the three simulation runs are carried out with the different seed numbers of 0.9, 183, and 3235 to generate the random numbers. The ‘Actual Centroid’ is calculated by taking the mean value of all $N$ samples and marked at 9.9837 for Seed (Sd) = 0.9, at 10.0176 for Sd = 183 and 9.9562 for Sd = 3235 in Figure 5(a, b and c) respectively and they are mentioned in Table 2 row: N-Mean (‘Actual Centroid’). The ‘Actual measurements’ are also shown in Figure 5 along with their measurement uncertainties ($10 \pm 1.5$ m) which are marked as ‘Expected Measurement Uncertainty lines’ for ‘Expected Centroid line’ and ‘Actual Measurement Uncertainty lines’ relating to ‘Actual Centroid’.

![Figure 5](image-url)
5.2.2 Fusion algorithms for removal of outlier information

In this step, the outlier samples are eliminated based on the calculated distance between ‘Actual Centroid’ and collected information. Based on sorted distance (minimum to maximum values), the k-number of nearest neighbors (kNN) to ‘Actual Centroid’ can be chosen with ‘k’ as the design choice.

This simulation is carried out with k = 10, 25, 50, 100, 250, 500, and 750 for all three seed numbers such as Sd = 0.9, Sd = 183, and Sd = 3235. However, Figure 5 shows only for k = 100 and the entire simulation summary is included in Table 2. The number of valid measurements per train is marked just below the ‘Actual Centroid’ line in Figure 5. The number of possibly chosen kNN measurements for k = 100 based ‘Expected Centroid’ (ideal value = 10) is shown just above the ‘Expected Centroid’ line.

In Table 2, ‘Single NN’ is named for k = 1 case. The ‘kNN-Mean’ row of Table 2 is computed based on the mean value of all kNN samples. In the case of ‘kNN-Prob’ values, all kNN samples are fused using its distance from the centroid as the probability. The probability of each sample is calculated in such a way that the lesser distance is given more weightage or vice versa [48].

As shown in Figure 5(a), for Sd = 0.9, train-5 and train-1 produce the greatest number of closest measurements around the ‘Actual Centroid’ line with the count of 15 and 13 respectively. Similarly, for Sd = 183, train-1 and train-6, according to Figure 5(b), and with Sd = 3235, train-10 and train-11, as seen in Figure 5(c), produce the greatest number of closest measurements around the ‘Actual Centroid’ line.

From the Table 2, it is observed that the best fusion accuracy with the least error of −0.0063 m is achieved by the ‘kNN-Mean’ algorithm with k = 500 for Sd = 0.9. Similarly, ‘kNN-Mean’ algorithm produces the least error of 0.0106 m with k = 750 for Sd = 183 and −0.0399 m with k = 250/50 for Sd = 3235. It is evident that ‘kNN-Mean’ with k > 250 and k < 750 shows better results compared to ‘N-Mean’. The results of ‘kNN-Prob’ are inferior compared to ‘kNN-Mean’. Interestingly, with k = 10 only, the results of ‘kNN-Prob’ are almost closer to “N-Mean” Overall, the simulation results confirm that the fusion accuracy of 1–4 cm is possible to achieve for the collected information.

Hence, depending on the number of samples available and accuracy needs, the choice of ‘kNN-Prob’ or ‘kNN-Mean’ can be made in the context of information fusion. From this case study, it can be concluded that for the least value of ‘k’, the ‘kNN-Prob’ algorithm is enough. With the larger value of ‘k’, it is essential to use the ‘kNN-Mean’ algorithm to achieve the best accuracy.

5.3 Case-3: RailML blockchain update

The objective of this case study is to show numerically how the inclusion of hash of hashes translates the typical RailML database into railML blockchain database (Step 4). For this, the Merkle root node of a railML is chosen as “railML.infrastructure”. The element “railML.infrastructure” contains the hash (SHA-1) of the number of “tracks.track” leaf nodes. As detailed in Figure 6(a), “railML.infrastructure@hash” attribute is calculated using the hash values of all the tracks, for example, tracks tr_01 and tr_02 along with value of infrastructure@id attribute. The hash of track id = “tr_01” is calculated as 705b7b948f22de2ca389d95e7d0d69b363d85 and the hash of track id = “tr_02” is calculated as 941bae3a1471cf43e532d4621d550d6e8cbf06h. The “railML.infrastructure@hash” = e5c7bb142eab734f87c48bd6ae46b6bec85382 is computed using the hashes of “tr_01” and “tr_02” along with “railML.infrastructure@id” = “i_01” as shown in Figure 6(a).

As described in Figure 6(b), each “railML.infrastructure.tracks.track@hash” is the SHA-1 of the combined attributes “track.trackTopology@hash”, “track.trackElements@hash”, “track.oscElements@hash” along with their separator “|”. The attribute “track.trackTopology@hash” of 87e471b1e954c993b77311 f31d 11b63510e6971 is calculated as the hash of hashes of “track.trackBegin”, “track.trackEnd”, and “track.connections” elements i.e. ece3f3a784464ce02fe6ebd2c37587769b10f, 344c6864f91f40ce3de7bf7f6bace 55127 and 615f759a8a38ee3c316ee342c5edc2e8b4b8d respectively. Similarly, the hash value of “railML.infrastructure.tracks.track.trackElements” element and “railML.infrastructure.tracks.track.oscElements” element are calculated as mentioned in Appendix-A.2.
5.4 Case-4: Permissioned blockchain-enabled information validation

The objective of this case study is to demonstrate numerically the substeps involved in the (a) transaction of collected information to be validated from a centralized server to another group of trains (train \( n+1 \) to \( 2n \)), (b) transaction of the validation results from another group of trains to the centralized server as compliant with permissioned blockchain. As illustrated in Figures 2 and 3(b), (downward arrow), at first, the centralized server signs the information to be validated using its \( PrK \) and broadcasts the same (Step 5). This information is extracted from the railML blockchain database as shown in Figures 3(b), 6, A.2. Other groups of trains receive this blockchain transaction (Step 6). Each train has to verify whether the received information is correct/authentic or not by using the public key of the centralized server (Step 7).

The steps involved in this case study is as similar to Section 5.1 except that the \( PrK \) and \( ShK \) of the centralized server is used. The blockchain transaction is performed between the centralized server and the specific train. Finally, the steps involved in Section 5.1 section are again repeated by the train to send the “validation” status message. The details are included in Appendix-A.3.

Once the collected information validation is confirmed through crowdsensing, it can be declared that the signalling

---

**FIGURE 6**  Typical railML blockchain database (a) “railML.infrastructure” element (b) “railML.infrastructure, tracks” with expanded track.trackTopology element

(a)

(b)
information is transformed into reliable data successfully. Then, the further steps on data design, data validation, and data verification can be executed (Steps 12, 13).

6 | CONCLUSION

A novel combination of railway vehicular crowdsensing-based trackside information collection and validation, information transaction over the permissioned blockchain layer between the train and the trackside centralized server, railML blockchain raw database, information fusion, and railML blockchain database is studied in the context of railway signalling data preparation process for the first time. Instead of manual, sequential, conventional signalling data preparation, real-time, automated, concurrent data preparation is proposed to overcome challenges. To enable concurrent information integration from in-service multiple trains and also to ensure information storage security, the blockchain-enabled railML databases are included in this paper. For the fusion of multi-train collected information, the nearest neighbourhood, and probabilistic data association-based algorithms are examined. To ensure the crowd and information trustworthiness and information transaction security, a permissioned blockchain-enabled transaction in crowdsensing information exchange mechanisms between the train and centralized server is used. Finally, the case studies are carried out to confirm the feasibility of implementing this proposed method within reasonable accuracy. The simulation results confirm the probable adaptation of the proposed method in performing secured trackside infrastructure information collection in the context of future ERTMS or PTC solution deployments over the tens of thousands of kilometres, of the railway lines around the world. As a future direction, the addition of more elements and attributes into railML blockchain databases can be considered. Also, several other information fusion methods can be evaluated in place of the discussed algorithms.

ORCID
Ganesan Muniandi 🐻 https://orcid.org/0000-0003-4681-0167

REFERENCES
1. European Train Control System. https://en.wikipedia.org/wiki/European_Train_Control_System (accessed on 17 May 2020)
2. Positive Train Control. http://stids.c4i.gmu.edu/papers/STIDS_Presentations/STIDS2015-Keystone1-MarkHartong.pdf (accessed on 17 May 2020)
3. Jean-Louis, B.: CENELEC 50128 and IEC 62279 Standards. John Wiley & Sons, New Jersey. pp. 155–200 (2015)
4. Schema railML.xsd. https://www.railml.org/files/download/schemas/2016/railML-2.3/documentation/railML.html#link429 (accessed on 17 May 2020)
5. railML version 2. https://www.clearsy.com/wp-content/uploads/2018/09/Formal-Data-Validation-CLEARSY-DATA-SOLVER-TOOL.pdf (accessed on 17 May 2020)
6. Van Gulijk, C., Hughes, P., Figueres-Esteban, M.: The case for IT transformation and big data for safety risk management on the GB railways. Proc. Inst. Mech. Eng., Part O: J. Risk Reliab. 232(2), 151–163 (2018)
7. The foundation for an universal infrastructure Data Exchange format. https://uic.org/IMG/pdf/160913_ermworkgroup_presentationerimpresentationparis170913.pdf (accessed on 17 May 2020)
8. Formal Data Validation, CLEARSY DATA SOLVER TOOL. https://www.clearsy.com/wp-content/uploads/2018/09/Formal-Data-Validation-august-2018.pdf (accessed on 17 May 2020)
9. Colominas, I., Molina, P.: Unmanned aerial systems for photogrammetry and remote sensing: A review. J. Photogramm. Remote Sens. 92, 79–97 (2014)
10. SigmaRail: Using drones to map the Spanish railways. https://www.railway-technology.com/features/sigmairail-using-drones-map-spanish-railways/ (accessed on 17 May 2020)
11. The future of rail surveying. http://www.railtechnologymagazine.com/Comment/the-future-of-rail-surveying (accessed on 17 Sep 2020)
12. Kwoczynska, B., Sagan, W., Dziura, K.: Elaboration and modeling of the railway infrastructure using data from airborne and mobile laser scanning. In: Baltic Geodetic Congress (BGC Geometrics). Gdansk, pp. 106–115 (2016)
13. GPSinftradat. https://www.bahnkonzept.de/en/compendium (accessed on 17 May 2020)
14. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: Current state and future challenges. IEEE Commun. Mag. 49(11), 32–39 (2011)
15. Ni, J., et al.: Security, privacy, and fairness in fog-based vehicular crowdsensing. IEEE Commun. Mag. 55(6), 146–152 (2017)
16. Crowdsourced Geospatial Data (CGD) – A Report on the Emerging Phenomena of Crowdsourced and User-Generated Geospatial Data. https://apps.dtic.mil/dtic/tr/fulltext/u2/a576607.pdf (accessed on 17 May 2020)
17. Misra, A., et al.: Crowdsourcing and its application to transportation data collection and management. Transp. Res. Record 2414(2014), 1–8 (2014)
18. Christin, D., et al.: A survey on privacy in mobile participatory sensing applications. J. Syst. Software 84(11), 1928–1946 (2011)
19. He, J., et al.: Towards mobile sensor-aware crowdsourcing: Architecture, opportunities and challenges. In: International Conference on Database Systems for Advanced Applications, Bali, Indonesia. pp. 403–412 (2014)
20. Muntazir, M., et al.: Referenceable mobile crowdsensing architecture: A healthcare use case. Procedia Comput. Sci. 134, 445–451 (2018)
21. Louza, M., et al.: Mobile crowd sensing architectural frameworks: A comprehensive survey. In: 7th International Conference on Information, Intelligence, Systems & Applications (IISA). Chalkidiki. pp. 1–7 (2016)
22. Hexagon selected for UK railway infrastructure mapping research project. https://www.thegeospatial.in/hexagon-selected-for-uk-railway-infrastructure-mapping-research-project-1424 (accessed on 17 Sep 2020)
23. Reyna, A., et al.: On blockchain and its integration with IoT Challenges and opportunities. Future Gener. Comput. Syst. 88, 173–190 (2018)
24. Blockchain for the IoT: opportunities and challenges. https://arxiv.org/abs/1805.02818 (accessed on 17 May 2020)
25. Ali, M.S., et al.: Applications of blockchains in the Internet of Things: A comprehensive survey. IEEE Commun. Surv. Tutorials 21(2), 1676–1717 (2019)
26. Blockchain: The Wheels Of Future Railway. http://www.businessworld.in/article/Blockchain-The-Wheels-of-Future-Railway/05-04-2020-168924/ (accessed on 17 May 2020)
27. Ali Syed, T., et al.: A comparative analysis of blockchain architecture and its applications: Problems and recommendations. IEEE Access 7, 176838–176869 (2019), https://doi.org/10.1109/ACCESS.2019.2957660.
28. Saxena, S., et al.: Blockchain based transactive energy systems for voltage regulation in active distribution networks. IET Smart Grid 3, 646–656 (2020)
29. The ERTMS/ETCS signalling system: http://www.railwaysignalling.eu/wp-content/uploads/2016/09/ERTMS_ETCS_signalling_system_revF.pdf (accessed on 17 Sep 2020)
30. Lopez, I., Aguado, M.: Cyber security study of the European Rail Traffic Management System. In: ERTMS Conference 2016A. Brussels (2016), DOI: 10.13140/RG.2.1.4832.8084
31. How Blockchain Addresses Public Key Infrastructure Shortcomings. https://www.infosecurity-magazine.com/opinions/blockchain-pki-shortcomings-1-1/#text=Application%20design%20has%20changed%20dramatically%20since%20PKI%20emerged%20text=Since%20Blockchain%20runs%20on%20tens%20of%20thousands%20of%20transactions%20(accessed%20on%2017%20September%202020)
32. Dai, H., Zheng, Z., Zhang, Y.: Blockchain for Internet of Things: A Survey. IEEE Internet Things J. 6(5), 8076–8094 (2019).
33. Ganesan, M.: Blockchain enabled virtual coupling of automatic train operation fitted mainline trains for railway traffic conflict control. IET Intell. Transp. Syst. 14(6), 611–619 (2020).
34. Multichain: https://www.multichain.com (accessed on 17 Sep 2020).
35. Juliette, M., Julie, B., Marion, B.: A Survey of GNSS-based research and developments for the European railway signaling. IEEE Trans. Intell. Transp. Syst. 10(18), 2602–2618 (2017).
36. Chen, Q., et al.: A Railway track geometry measuring trolley system based on aided INS. Sensors (Basel) 18(2), 1–26 (2018).
37. Trackside railway apparatus and method for detecting use of at least one trackside component of a railway installation. https://patents.google.com/patent/WO2017008078A1/en (accessed on 17 May 2020).
38. Truth Discovery for Passive and Active Crowdsourcing. https://cse.buffalo.edu/~jing/doc/CIKM16.pdf (accessed on 17 May 2020).
39. Estrin, D.: Participatory sensing: Applications and architecture [Internet Predictions]. IEEE Internet Comput. 14(1), 12–42 (2010).
40. Ganesan, M.: Blockchain enabled virtual coupling of automatic train operation fitted mainline trains for railway traffic conflict control. IET Intell. Transp. Syst. 14(6), 611–619 (2020).
41. Choodowski, E., Lisiecki, P., Lech, P.: Hybrid algorithm for the detection and recognition of railway signs. In: CORES: International Conference on Computer Recognition Systems. Polanica-Zdroj, Poland. pp 337–347 (2019).
42. Railway track specific traffic signal selection using deep learning. https://arxiv.org/abs/1712.06107 (accessed on 17 May 2020).
43. Karagiannis, G., Olsen, S., Pedersen, K.: Deep learning for detection of railway signs and signals. In: Computer Vision Conference. Las Vegas, NV. vol. 943, pp 1–15 (2019).
44. Marjanovic, M., Antonic, A., Zarko, I.P.: Edge computing architecture for mobile crowdsensing. IEEE Access 6, 10662–10674 (2018).
45. ECDSA: Elliptic Curve Signatures. https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages (accessed on 17 May 2020).
46. The Corda Platform: An Introduction. https://www.corda.net/content/corda-platform-whitepaper.pdf (accessed on 17 Sep 2020).
47. Address and key format. https://www.multichain.com/developers/address-key-format/ (accessed on 17 Sep 2020).
48. Ganesan, M., Ezhilarasi, D.: Train distance and speed estimation using multi sensor data fusion. IET Radar Sonar Navig. 13(4), 664–671 (2019).

How to cite this article: Muniandi, G.: Blockchain-enabled secure crowdsensing for trackside infrastructure information collection and validation in railway signalling data preparation. IET Blockchain. 1, 16–32 (2021). https://doi.org/10.1049/blic.2.12002

APPENDICES

A.1 | Description of railML blockchain database

RailML blockchain database stores trackside railway signalling information and configuration data at each railway infrastructure area. As detailed in Figure A.1, the railML blockchain XML database contains the root node as “railML” which has the key element named, “railML.infrastructure”. The element node “railML.infrastructure” acts as the Merkle root node of a railML which will contain the number of “tracks.track” leaf nodes. Each “railML.infrastructure.tracks.track” element contains the number of attributes such as “track@id”, “track@type”, “track@hash”.

The child elements of “track.trackTopology”, “track.trackElements”, “track.ocsElements” contain by default four attributes such as @hash, @id, @pos, and @absPos. The attribute @hash stores the calculated SHA-1 value as per the comment “< !–Hash for” as mentioned inside the railML file as shown in Figures 6, A.2. The attribute @id is used for the unique identification of any railML trackside infrastructure object. The attributes @pos and @absPos represent the relative distance from the beginning of the same track and absolute location in the track respectively.

The element “track.trackTopology” defines the key information which is related to the beginning and end of the track via “track.trackBegin” and “track.trackEnd” elements. This information is either calculated at the first step before the start of actual information collection or fed manually using some tools or collected manually through the crowdsensing technique.

The element “track.trackElements” contains the information which majorly depends on track geometry, for example, “speedChanges”, “gradientChanges”, “radiusChanges”, “levelCrossing”, “platformEdges”, “tunnel”.

The element track.trackElements.speedChanges provides the speed restriction value “track.trackElements.speedChanges.speedChange@id” which has to be followed by the train from the location “track.trackElements.speedChanges.speedChange@pos”.

The element track.trackElements.gradientChanges provides the slope “track.trackElements.gradientChanges.gradientChange@slope” of track section starting at “track.trackElements.gradientChanges.gradientChange@id”.

Similarly, the element track.trackElements.radiusChanges provides the change in radius of curvature “track.trackElements.radiusChanges.radiusChange@id” of track section starting from “track.trackElements.radiusChanges.radiusChange@pos”.

The elements “track.trackElements.levelCrossing”, “track.trackElements.platformEdges”, “track.trackElements.tunnel” provide the information about the location of corresponding elements.

The element “track.ocsElements” contains the information related to railway operation and control system objects, for example, “signals”, “trainDetectionElements”, “balises” etc.

The elements track.ocsElements.signals, “track.ocsElements.trainDetectionElements”, “track.ocsElements.balises” provide the information about the location of corresponding elements. In addition, the attribute @dir is included in the elements “track.ocsElements.signals.signal” and “track.ocsElements.balises.balise”. More detailed information is present in [4].

A.2 | Description of hash calculation for “trackElements” and “ocsElements”

As indicated in Figure A.2a, the element “trackElements@hash” contains the information which majorly depends on track geometry, for example “speedChange@id = ’sc_01’, “gradientChange@id = ’gc_01’, “radiusChange@id = ’rc_01’, “levelCrossing@id = ’lc_01’, “platformEdge@id = ’pl_01’, “tunnel@id = ’tl_01’. The hash value of “trackElements@hash” is computed as the
As detailed in Figure A.2b, the “signal@hash” = b327c60abb92671897f1b28d7f77e5d935f6c which is determined using the attribute values, id = “sig_01” pos = ‘0’ absPos = ‘1000’ dir = “up”. Similarly, the “trackCircuitBorder@hash” is estimated using the attribute values, id = “tc_01” pos = ‘0’ absPos = ‘1000’ as 9e550eb75f1f5276e549b16e2b5a547b6c6e78d. In the same way, the “balise@hash” is calculated as 3b275892db87b65f67bc87f7f7e0f1b using the attribute values, id = “ba_01” pos = ‘0’ absPos = ‘1000’ dir = “up”.

A.3 | Permissioned blockchain-enabled information validation

For information validation, also, the Elliptic Curve Digital Signature Algorithm (ECDSA) is used by the centralized server (Step 6), as similar to permissioned blockchain, for example, Multichain [47]. For example, the PrK (256 bits, 64 hexadecimal digits) of the centralized server is assumed as, 

\[ Prk = 5B2D3F900AB230E4714F67D9F6E8792E43B39BE89FC9BD3D20EAC29DC0A4F616 \]

The ShK (512 bits, 128 hexadecimal digits + ‘04’ prefix) of the centralized server is derived from PrK as,
As an example, the fused information about the signal in ‘up’ direction along with the track id = tr_01 of infrastructure id = i_01 at the distance of 10.08 m is assumed as,

Information for Signal = ‘Signal|i_01|tr_01|10.08| up’

For the blockchain transaction, the hash of the signal information is calculated using the double SHA-256 algorithm first and then, signed using PrK of the centralized server as,

\[ \text{Signature} = \text{ECDAA_Sign}(\text{SHA-256}(\text{SHA-256}(\text{Information})), \text{PrK}) \]

The SHA-256 of the signal-1 information from the centralized server is determined as,
\[ \text{SHA-256(Data)} = 57C179266146634BB64C79806347C77821E3478A932150F3B30EF2E95E1ACBC \]

The signature of the centralized server information is formulated using “SHA256withECDSA” as, \( \text{Signature} = 3046022100B0075C33159416446207A5053AD0EF6 \)
\( \text{CBC788E92B35D5357494FB8779696CFDB022100B577A4FD3E03AF34358BB63A53762D7B9F45BEFCF3F4537ABA130903B111E3B22} \). This signature can also be transmitted in the encoded form.

The verification of proper reception using the public key (Step 7) is also carried out in two substeps: First, the hash of the information is recreated using SHA-256. Then, the signature verification is performed using ShK of the centralized server as,

\[ \text{Verification} = \text{ECDSA\_Verify(SHA-256(SHA-256(information)), Signature, ShK)} \]

If the verification result is ‘true’ then the blockchain transaction is declared as ‘valid’. Then, the information is split using “|” separator, and the Infrastructure type is identified as ‘Signal’. In the end, the infrastructure id and track id are extracted. Each train will look for the @pos attribute value of the signal to confirm its physical presence on the track.

Finally, the steps involved in Section 5.1 section are again repeated by the train to send the “validation” status message, i.e. Pass (1) or Fail (0) as indicated in Figure 3b (upward arrow, Steps 9, 10, 11).

On its reception, the centralized server updates the railML blockchain database with the validation status appropriately by updating the attributes such as \text{signal@valPassTrainIds} and \text{signal@valFailTrainIds}, using the train identification number, for example, \(<\text{signal hash} = \text{d01f7878b487fc6402e37582af7e1298ea3d3c} \text{time} = \text{2020-04-24:09-05} \text{pos = } 10.08 \text{dir = “up” valStatus = “1-Pass” valPassTrainIds = “train-21;train-22;train-23” valFailTrainIds = “train-24”} > \). Lastly, the centralized server updates the overall validation status attribute i.e. \text{signal@valStatus} as ‘Pass’ when the difference between the number of trains @ \text{valPassTrainIds} and @\text{valFailTrainIds} > 5 (which can be a design choice).