Article

A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph

Yongbae Kim ¹, Juyong Back ² and Jongweon Kim ^{1,3,*} ¹ Creative Content Labs, Sangmyung University, Seoul 03016, Korea; ybkimdb@gmail.com² Quantum Gate Inc., Sejong 30043, Korea; v@quantumgate.co.kr³ Department of Electronics Engineering, Sangmyung University, Seoul 03016, Korea

* Correspondence: jwkim@smu.ac.kr

Abstract: A tachograph in a vehicle records the vehicle operating conditions, such as speed, distance, brake operation conditions, acceleration, GPS information, etc., in intervals of one second. For accidents, the tachograph records information, such as the acceleration and direction of a vehicle traveling in intervals of 1/100 s for 10 s before and after the accident occurs as collision data. A vehicle equipped with a tachograph is obliged to upload operation data to administrative organizations periodically via other auxiliary storage devices like a USB attached external memory or online wireless communication. If there is a problem with the recorded contents, data may be at risk of being tampered with during the uploading process. This research proposed tamper-resistant technology based on blockchain for data in online and offline environments. The suggested algorithm proposed a new data recording mechanism that operates in low-level hardware of digital tachographs for tamper-resistance in light blockchains and on/offline situations. The average encoding time of the proposed light blockchain was 1.85 ms/Mb, while the average decoding time was 1.65 ms/Mb. With the outliers in statistical tests removed, the estimated average encoding and decoding time was 1.32 ms/Mb and 1.29 ms/Mb, respectively, and the tamper verification test detected all the tampered data.

Citation: Kim, Y.; Back, J.; Kim, J. A Tamper-Resistant Algorithm Using Blockchain for the Digital Tachograph. *Electronics* **2021**, *10*, 581. <https://doi.org/10.3390/electronics10050581>

Academic Editor: Andres Marin

Received: 16 January 2021

Accepted: 26 February 2021

Published: 2 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: digital tachograph (DTG); blockchain; tamper resistant

1. Introduction

Today, vehicles are a means of transport that has provided mobility for human activities and has brought groundbreaking changes to passenger transportation and logistics. With the advancement of technology, improvement in vehicle transportability and driving speed plays a vital role in economic development, but is a cause of serious accidents that lead to human injuries and deaths. Vehicle operation by human control is inevitable until autonomous driving technology reaches the completion stage, so it is impossible to eliminate the occurrence of traffic accidents. Therefore, all countries around the world plan to reduce accidents through regulations related to vehicle operation. However, there are limits to alleviating traffic accidents by regulations alone. For supplementation, diversified efforts are being made to reduce traffic accidents by mandating tachograph installation for vehicles to inspect long-term driving without rest or violation of traffic safety regulations, such as reckless driving.

Figure 1 shows the number of deaths per 1,000,000 people related to traffic accidents among OECD countries in 2018 [1]. The death toll is high in the United States, as it has the highest number of automobiles being operated, and in Chile and Turkey, where traffic control is relatively less systematic. The number of deaths in Korea is the seventh highest among OECD countries even though the traffic is well controlled due to shortcomings in the penalties by the Road Traffic Acts and the systems related to the usage of tachographs. Through the improvement of the Road Traffic Acts, such as strengthening the penalties against driving under the influence, the number of deaths per 1,000,000 people decreased

to 64.77 people according to the 2019 statistics, but still exceeds the average among OECD countries in 2018 (approx. 53.1 people).

Figure 1. Number of deaths in traffic accidents per 1,000,000 population in OECD countries.

The European Union [2] has mandated installing digital tachographs or smart tachographs to record a driver's occupational behavior, such as rest time and operation time. According to EU Social Norm No. 561/2006, it provides reliable data to administrative executors to increase road safety, guarantee working conditions for occupational drivers, and seek fair competition among countries within the EU. In December 2015, the United States FMCSA (Federal Motor Carrier Safety Administration) enacted a law to make the ELD (Electronic Logging Device) mandatory for all commercial vehicles [3], which has become effective as of 18 December 2017. According to this law, a truck driver's working hours are limited to 14 h a day, where 11 h are permitted for actual driving and the remaining 3 h are for loading and resting.

In Korea, "Management Guidelines for Automobile Operation Records and Devices" [4] were announced, according to the Traffic Safety Act Article 55, Enforcement Ordinance Article 45 of the same Act, and Enforcement Regulation Article 29 and 30 of the same Act for efficient implementation of business related to storage, submission, inspection, analysis, and utilization of operation records. Installation of tachographs has become mandatory for all trucks (commercial vehicles according to Traffic Safety Act Article 55) since 2005 and for newly registered vehicles of 1 ton and above for cargo transportation business since 2011. The purpose of digital tachograph installation is to reduce the occurrence of accidents; therefore, post-inspection is conducted to prevent reckless

driving and encourage safe driving, which, as a result, has the function of preventing accidents. For digital tachographs currently in operation, it is mandatory to periodically upload operation records to administrative organizations (Traffic Safety Act, Article 55, Paragraph 2). However, as of 2016, there were 97,322 buses, 254,909 taxis, 252,729 cargo trucks out of commercial vehicles that were mandatorily equipped with tachographs, but only 89.9% of buses, 33.9% of taxis, and 17.6% of cargo trucks submitted the records of tachographs [5]. Cargo trucks that rank the highest among highway traffic accidents are the most uncooperative in submitting operation records. Legal loopholes, such as no sanctions on such non-submission or no utilization of submitted operation records for illegal driving control, are being revealed.

The submission rate of operation records is low, except for buses because non-digital tachographs do not perform the upload function properly. The inconvenience of current tachographs is that the data needs to be copied on an SD (Secure Digital) card or USB attached external memory and then use an exclusive program on a computer connected to the internet for upload. However, a more severe issue is that the records are defenselessly exposed to tampering, while accessing and uploading operation records stored on SD cards or USB attached external devices. Therefore, functions for tampering prevention or check on digital tachographs is essential.

The technology used in actual digital tachographs supports fast booting and prompt data storage using SLC (Single Level Cell) NAND flash memory to store data with hardware resources securely [6], while Park implemented low cost and high-speed performance for tamper resistance using monotonic counters [7]. However, the performance of ADAS (Advanced Driver Assistance System) devices are enhancing rapidly due to the development of autonomous driving technology. There was a study on the enhancement of driving safety using ADAS information for vehicle operation recording functions [8], as well as a study on the intelligent ADAS system that assists the driver by providing information on the distance from the vehicle ahead of you and equipped with accident video transmission function that automatically stores and transmits videos of the accident in real-time for the efficient control of the accident, prevention of additional accidents in advance, and accurate investigation of the accident [9]. Such changes in the environment are affecting the changes in tamper-resistant technology. With the recent development of blockchain technology, there are increasing demands to use blockchains for data tampering prevention. However, the computing resource utilization would be too high to execute blockchain in low-level hardware such as tachographs, so this issue needs to be addressed in advance.

While the adoption of digital tachographs for commercial vehicles is changing into a global trend, there is no guarantee of the integrity of the data for stored driving records. Drivers can try to forge their own driving record data in the process of transferring data using an SD card or USB attached external memory, or while offline. In this study, with a blockchain-based algorithm, it is possible to effectively prevent forgery and falsification of data, even in an offline state without a network connection or forgery during data transmission by an external storage device. Additionally, we developed the light blockchain algorithm that can minimize computation while using the basic characteristics of blockchain that is excellent for tamper resistance of operation data in tachographs and conducted the tamper-resistance and lightweight performance verification for actual operation record data.

2. Related Works

2.1. Blockchain

Blockchain technology is the representative technology in chain algorithms. A chain algorithm encodes numerous transactions and connects them in chains for tamper-resistant, and decentralized dispersed storage for data management. A chain refers to the connection of multiple chains where the previous block's hash becomes a component of the next block. The data structure connected by chains is similar to the linked list method. Multiple nodes that comprise a database are aligned in chronological order, and are connected in

chain format. The existing linked list data structure had separate pointers for marking the previous node's address, and the following nodes' address. However, in a blockchain data structure, the hash itself takes the role of the address, which connects each node.

Blockchain first attracted attention from the world when someone under the pseudonym Nakamoto [10] proposed developing a cryptocurrency called Bitcoin in November 2008 and developed the bitcoin based on the blockchain algorithm in the following year (January 2009). A blockchain is an algorithm that binds multiple transactions to comprise a single block and connects multiple blocks in the form of a chain using a hash, which is copied and stored in dispersion. Blockchain technology is tamper-resistant and enables decentralized and diversified management, which was a revolutionary change compared to previous data management methods. Blockchain technology was first applied to bitcoin, followed by many cryptocurrencies, including Litecoin [11], Ethereum [12], Ripple [13], Bitcoin Cash [14], and EOS [15].

Despite many advantages of blockchain, it has a serious disadvantage, which is "very slow processing speed". If a transaction occurs, it cannot be processed instantly, but has to wait until numerous transactions are collected to comprise a block. Even after a block is composed, there is a long waiting time to verify and examine multiple nodes dispersed throughout the network. For example, in the case of a Bitcoin, it takes about 10 min to compose a new block, and often it takes more than one hour for that block to be examined on the network. Thus, it is impossible to use cryptocurrencies such as Bitcoins based on blockchain as an everyday payment method. Bitcoins and other cryptocurrencies can be used as a value storage method, but are difficult to use as a payment method in everyday life, because of the limits in the blockchain algorithm itself.

Each of the blocks that form a blockchain is fixed in size, so scalability is becoming an issue. At first, when there were not many blockchain users, the limits in block size were not a big issue. However, as the number of blockchain users increased tremendously, the data that a single block can store often exceeds its maximum limits. For example, in the case of Bitcoins, the maximum size of a single block is restricted to 2 Mb. Therefore, if approximately 2000 transactions are recorded per block, there is not enough space to record more transactions. In such a case, the relevant transactions are not recorded on the block and are pushed back. Of course, you can pay a higher fee to move it to the front to record it on the block first, but this can lead to an inflation of fees. A temporary measure, called SegWit, was applied as of 1 August 2017, to save more hash data on each block by excluding the user's digital signature from being recorded on the block. However, the fundamental issues remain unsolved. There are suggestions that the block size should be continuously increased to 4 Mb or 8 Mb, yet it may cause a waste of unused space, which leads to the entire blockchain becoming heavier and slower.

2.2. Improvement of Blockchain

New algorithms with some functional improvements are in development to resolve the slow speed and scalability issues of blockchain. J. Poon and T. Dryja suggested a lightning network [16] to resolve the slow speed of existing blockchains and realize speed as fast as lightning. The suggested algorithm processes individual transactions in separate channels and only records the results on the blockchain. J. Poon and V. Buterin developed a plasma algorithm that uses a similar method. In this algorithm, all transactions are processed in a separate child chain instead of the main chain, and only the results are transmitted to the main chain. G. Konstantopoulos improved the plasma and proposed the plasma cash algorithm [17]. In the plasma cash algorithm, users do not download all the blocks for verification as they used to in the existing algorithms, but track only the blocks that include specific coins that individual users show interest in to enhance the processing speed. Due to the launch of these new algorithms, it is anticipated that the slow speed of existing blockchains will be improved to a faster speed.

Meanwhile, in the case of newly developed cryptocurrencies, in the initial stage of its introduction, the number of users participating in the network is remarkably small, so

vulnerability to data tampering may arise. To defend the so-called 51% attack, it could be more advantageous for new coins to add their transactions to an existing blockchain that many users are already participating instead of having independent blockchains of their own. Such a chain is called a side chain [18]. In a side chain, Ethereum, QTUM [19], Nem (New Economy Movement) [20], and Hdac [21], which are platform coins, are adjacent to the main chain and operate by adding its node to the existing main chain's node.

2.3. Light Blockchain and Transportation

Lightweight blockchains are an essential topic for research in the field of the Internet of Things (IoT) to apply the concept of computing in computing environments with limited resources. As too many resources and power consumption are required to use blockchain in the industrial Internet of Things (IIoT) devices, researchers, including the researchers in Reference [22], proposed separating full node and light node and processing IIoT devices in light node. In light node, a lightweight data structure called LightBlock and a blockchain system called LightChain were proposed. L. Bai et al. [23] proposed a lightweight blockchain network architecture. This network separates an on-chain network that handles processes related to blockchain and an off-chain network that handles processes that cannot be resolved by blockchain to lighten the blockchain. D. Gruber et al. [24] distributed filters for lightweight clients to provide a fair environment for provisioning. T. Lin et al. [25] proposed an improved blockchain sum algorithm based on a mortgage model instead of a probability model, intersecting chain protocol with horizontal diffused capacity and high-performance cross-chain blockchain network structure that processes more than 1000 transactions per second through verification.

Blockchains may promote the establishment of a safe, reliable, dispersed intelligent transport ecosystem to solve the data-sharing issue, thus contributing to the better use of transport infrastructure and resources [26–28]. Singh and Kim proposed an intelligent vehicle trust bit mechanism that supports the security network between vehicles using blockchain, and J. Kang et al. [29] studied the safe and efficient data sharing on vehicular edge computing using blockchain technology. A. Didouh et al. [30] used a consortium-type blockchain for electronic toll collection, and featured a GPS counterfeit prevention function using a radio base station on the road. S. Lee et al. [31] wanted to use a public blockchain network as a platform that can provide driver information, vehicle information, and traffic information of vehicles. B. Bera et al. [32] built a drone operating system that can process a lot of data in a 5G IOT environment using a private blockchain. G. Saldamli et al. [33] used Ethereum-based blockchain to provide reliable car information to insurance institutions, vehicle dealers, and institutions. A. Lei et al. [34] proposed a system that exchanges information safely and effectively using a decentralized public blockchain in the Heterogeneous Intelligent Transportation System (ITS).

Several researchers have attempted to apply the blockchain to the transportation system, but no research has been conducted to prevent forgery and alteration for the operation data of digital tachographs. Additionally, they did not consider the situation where the device is not connected to the network.

3. Light Blockchain for Digital Tachograph

Tachographs installed on vehicles record various data related to the vehicle operation. The data can be used at any time to identify information related to safe driving, or the driver's resting conditions. In Korea, it is mandatory to store the operation record data of commercial cargo trucks for the previous six months and submit it if there is a request from the administrative organization. However, the data may be exposed to tampering while being moved to an SD card or USB attached external memory or transmitted via a computer for submission. Therefore, multifaceted efforts are required for tamper-resistance, and blockchains are an effective solution for that.

The slow speed, a major drawback of blockchain, is caused by the basic characteristics of blockchain and complicated arithmetic processing method. Various efforts are being

made to solve this issue; however, the high-performance computer (with GPU embedded) is the prerequisite for the system that utilizes the blockchain. Therefore, small devices such as IoT (Internet of Things) devices have limits in properly using the blockchain. The embedded hardware of the DTG (Digital Tachograph) also has a simple structure, which cannot be used for complex blockchain technology.

3.1. Outline of Light Blockchain Method

The blockchain encryption algorithm was used for tamper-resistance DTG. This algorithm, called the light blockchain algorithm, implements a blockchain algorithm suitable for mobile devices, taking into consideration the data processing capacity, calculation capacity, etc. The characteristics of light blockchain algorithm is that the device needs to be tamper-resistant even in conditions when it is not online, taking into consideration the characteristics of a mobile device. The system has been formed so that it will perform functions that are suitable for a mobile device.

The basic concept of the light blockchain is the same as the permissioned blockchain. However, there is no consensus algorithm in the light blockchain, and while the light blockchain generates blocks on the mobile terminal, the permissioned blockchain creates blocks on the orderer. In addition, the light blockchain can create blocks even in the on/offline state, but the permissioned blockchain can only create blocks in the online state. Table 1 shows the comparison of the characteristics.

Table 1. Comparison of blockchain characteristics.

Type	Permissioned Blockchain	Light Blockchain
Composition	Identical	Identical
Consensus algorithm	CFT (Crash Fault Tolerance)	None
Signature	CA (Certificate Authority)	Key value registered on server
Block generation	Block generated by orderer, and saved by peer	Block generated and saved on mobile device
Authentication	Electronic wallet using MSP (Membership Services Provider)	Electronic wallet, password login
Execution	Online connection on blockchain network	Online, offline operation
Application to mobile environment	Difficult to apply on mobile device, due to limits in calculation time and processing capacity.	Suitable to mobile environment. Purpose is fast execution speed and tamper-resistant.

3.2. Proposed Light Blockchain Algorithm

The proposed algorithm is based on a light blockchain structure that is tamper-resistant for operation record data on DTG. The proposed light blockchain uses a similar architecture to the existing blockchain, except a random hash. If the data of the block that is already formed is tampered with by using the hash value of a previous block, the data of all blocks need to be recalculated to calculate the hash value of the current block. This process requires authentication such as Proof of Work or Proof of Stake and makes tampering practically impossible. Figure 2 shows the structural differences between general blockchain and light blockchain. The Proof of Work on public blockchain uses Nonce value and closed blockchains include electronic signatures and endorsements for each transaction. In the proposed light blockchain, a random hash performs this function so that the block is tamper-resistant even when it is offline. Equation (1) is the formula of the light blockchain.

$$LB_{k+1}(BLK_{k+1}) = SHA128(LB_k(BLK_k) + SHA128(DataFile) + SHA128(PRN + SeedCount + BLK No)) \quad (1)$$

Figure 2. Light Blockchain Architecture.

Systems that use blockchains must always be online, but in a mobile environment such as in vehicles, various reasons may disconnect the communication. In this case, data is exposed to the threats of tampering. Figure 3 shows that block generation is possible even when communication is disconnected, by using random seeds. When block data is modified offline to regenerate a block, the block is generated normally on the device, and it is possible to check for tampering by comparing the difference of the ServerSeed value registered on the server with the number of calls on a random function of the regenerated block or with the number of block generation. In this paper, an anti-copying algorithm for tamper-resistant blockchains was implemented using random functions with ServerSeed provided by the server and BlockNumber SeedCount as variables. The ServerSeed and BlockNumber provided by the server generate a new Block Number each time a block increase. The SeedCount value is a variable that only increases when communication is disconnected. Therefore, if the data file is modified and blocks are regenerated, the BlockNumber ServerSeed value remains the same. In contrast, the SeedCount value changes each time a block is made, while communication is disconnected. These values are used to prevent tampering.

Figure 3. Block generation of On/Off-Line.

3.3. System Registration Process

Only users registered on the server can log on to the light blockchain system, so managers, devices, and users of a company are registered, as shown in Figure 4. An electronic wallet is created in the registration process, which can be used as a USB attached external storage or smartcard.

Figure 4. Registration process.

3.4. BlockHash Calculation during Online Operation

When the network is online, the block is generated through the following process. When the block data is ready, the ServerSeed is requested to the server, and the received ServerSeed value is used to calculate the ServerKey value. If the network is online and the server response is received, the SeedCount is reset and the ServerSeed is updated by the received ServerSeed value. The updated ServerSeed value and BlockNumber SeedCount value are used to calculate the ServerKey hash and BlockData hash value through a random function. The previous BlockHash value is used to calculate the BlockHash value. The calculated BlockHash value is transmitted to the server, and the result is recorded on the block header.

Figure 5 shows the online block generation process in brief. (1) On the DTG device, ServerSeed request is made by sending the Device ID, User ID, and Block Number. If the network is online, (2) the ServerSeed is allocated from the server using Device ID, User ID, and Block Number. The allocated ServerSeed is used (3) to calculate ServerKey. When the BlockHash is calculated, that value is used (4) to generate a block. When the block generation is complete, (5) the BlockHash value is transmitted to the server. When the transmission to the header is complete, the block status is set as (6) Transmission complete.

Figure 5. On-Line process of a block generation.

The online block generation process can be expressed by pseudo-codes in detail, as shown in Table 2.

Table 2. Pseudo Code for On-Line Process: Digital Tachograph.

```

1: for I = 1 to n do
2:   if server connection is available then
3:     ServerSeed  $\leftarrow$  Integer value received from server
4:     SeedCount  $\leftarrow$  0
5:     ServerKey  $\leftarrow$  Calculate random value with ServerSeed, BlockNumber, SeedCount
6:     ServerKeyHash  $\leftarrow$  Calculate hash of ServerKey
7:     BlockDataHash  $\leftarrow$  Calculate BlockData hash value
8:     BlockHash  $\leftarrow$  Calculate Merkle tree with ServerKeyHash, PreviousBlockHash,
       BlockDataHash
9:   endif
10:  if (BlockHash value can be transacted to the server) then
11:    BlockStatus  $\leftarrow$  "T" { Transaction okay}
12:  else
13:    BlockStatus  $\leftarrow$  "U" { Transaction fail }
14:  endif
15: endfor

```

3.5. Offline Operation

During offline operation, where the ServerSeed cannot be received due to communication unavailability, the previously received ServerSeed is used as is. The SeedCount is then activated instead to increase the value each time a block is created and internally perform the encryption process of the block. It is impossible to check for tampering within the device, and the internal algorithm recognizes it as a normal file. Suppose the block data contents are modified, and a block is regenerated when communication is unavailable. In that case, the block creation counter keeps operating internally to show differences with seed counters created by a normal block generation process. For this reason, tampering of blocks occur during communication unavailability can only be checked on the server.

Figure 6 shows the offline block generation process in brief. (1) On the device, the ServerSeed request is made. If the network is offline, (2) it cannot receive the ServerSeed from server. If the offline status is conformed, (3) the SeedCount value is changed. (4) The changed SeedCount and the previous ServerSeed (OldServerSeed) value is used to calculate a random variable (ServerKey), (5) and that result value is applied to the BlockHash calculation. (6) The BlockHash value, previous BlockHash value, and ServerKey hash values are used to connect a blockchain. (7) If it is still offline, the block status is set as non-transmission. The offline block generation process performs the prevention of block tampering when the server is not connected.

Figure 6. Off-Line process for a block generation.

The offline block generation process can be expressed by pseudo-codes in detail, as in Table 3.

Table 3. Pseudo Code for Off-Line Process: Digital Tachograph.

```

1: for I = 1 to n do
2:   if server connection is not available then
3:     ServerSeed  $\leftarrow$  Use the previous stage ServerSeed value
4:     SeedCount  $\leftarrow$  Increment SeedCount
5:     ServerKey  $\leftarrow$  Calculate random value with ServerSeed, BlockNumber, SeedCount
6:     ServerKeyHash  $\leftarrow$  Calculate hash of ServerKey
7:     BlockDataHash  $\leftarrow$  Calculate BlockData hash value
8:     BlockHash  $\leftarrow$  Calculate Merkle tree with ServerKeyHash, PreviousBlockHash,
       BlockDataHash
9:   endif
10:   BlockStatus  $\leftarrow$  "U" { Not Transaction to server }
11: endfor

```

3.6. Data File

The data file needs to be stored and maintained according to the format designated by the Ministry of Land, Infrastructure and Transport [4] at intervals of 1 s for operation information data, and 1/100 s when a collision is detected on the vehicle. The size of block data files are marked in bytes and displayed in nine digits; therefore, the maximum size does not exceed 999 Mb. When calculating the maximum operation record data [30], the data is generated at intervals of one second for 24 h operation. Hence, it is calculated as $24 \text{ (h)} \times 3600 \text{ (s)} \times 147 \text{ (byte)} = 12,700,800 \text{ bytes}$ and considered appropriate to display the file size in nine digits.

3.7. Light Blockchain Management File

Within the DTG device, an internal blockchain list is managed according to the Management Guidelines for Automobile Operation Records and Devices for tamper-resistant blockchain encryption and management. This file designates that the Genesis Block that is created when first registering the device as No. 00000, and each time a block increases, it is registered on the management file. The name of the management file is ".blocklist.dat" and is recorded in the same location as the execution file as a hidden file. The block composition of the light blockchain is shown in Table 4, and details on each item are as follow:

Table 4. Block Composition.

No.	Item	Form	Note
1	Current block number	00001	5 digits
2	Block creation date	20191112	9 digits
3	Data file name	0000120191111	13 digits
4	Data file size (byte)	000319877	9 digits
5	Previous block hash value	a48c1ec9412e5a66df09503b3cdcb4b8a16d085	sha128
6	Current block hash	43f3e0a4aa388e1eef519f55118f5be3400dd989	sha128
7	Current block status	D:Blockchain encryption, V:Verification of tampering, X:Tampered, T:Blockhash transmitted, U:Blockhash non-transmitted	1digit
8	Time and date of block creation	20191112 04:49:44	17

- Current block number

The block number is displayed in five-digit numbers from 00000 to 65535. The block first created on the system is called Genesis Block and is numbered 00000. Only one block is created per day, so it has enough space to record 180 years of data, even if it is operated every day.

- Block creation date

The block creation date is when the DTC device records data and is displayed in year, month, and day as follows:

20191112: 12 November 2019

- Data file name

It is a raw data file created internally within the DTG device and original data that forms a light blockchain. The name and location of the original data are as follows. The No. 00001 block is used to display the data created on 12 November 2019, and the data is saved in the directory with the data name under the program executed.

blockID+Date+.d: 0000120191112.d

blockID: It is a block number of light blockchains with consecutive numbers. The first block created is assigned No. 00000 and called a genesis block. Subsequently, a block is created continuously, increasing by one at a time.

Date: It indicates the date the vehicle was operated, and only one block is created per day. The format of indicating dates is as follows:

20191112: 12 November 2019

- Previous block hash value (PreviousBlockHash)

It is a block hash value calculated in the previous block, which corresponds to a connecting link to a blockchain. It is configured with 30-digit hexadecimal number data using the SHA1 hash algorithm.

- Current block hash value (BlockDataHash)

A DTG device has many restrictions in data processing capacity as a mobile device. Therefore, the SHA1 hash value was used as it is relatively easier on a load of calculations. For tamper-resistant encryption of data, a code value issued on the server and block number data was formed into a Merkle tree, as shown in Figure 7 for encryption, instead of using an electronic signature algorithm, for prompt verification of data tampering.

Figure 7. Merkle tree for light blockchain.

- ServerKeyHash value

Variables used for ServerKeyHash value calculation are ServerSeed, BlockNumber, and SeedCount. The ServerSeed, a 32-bit integer provided by the server, is a seed value that verifies the block hash value of the device on the server. The block number is the continuous value of the block that is created in units of one day. A block is only created on the days the vehicle is operated, not on the days that the vehicle is not operated. The SeedCount value is an encrypted file within the device and can only be opened by the User

Wallet. The seed counter is used for tamper-resistance when the communication status is unstable.

- Block status

The block status is displayed, as shown in Table 5. The block is flagged as D when Day 1 data block is completed in the DTG device and registered as a block on the light blockchain. The verified block is flagged as V. If a change is made to the block data for any reason and the verification program detects such a change, the changed block is flagged as X. If it is connected to the blockchain, the block hash value is transmitted to the server, disabling any tampering within the device. However, when the block hash value cannot be transmitted due to communication errors, the block status is registered as U, indicating a non-transmission status. Blocks registered as U transmits the registered block hash value to the block header when communication becomes available again. The transmission process takes place after checking the communication status of the vehicle that is not moving but with the engine running.

Table 5. Block Status Flag.

Block Status	Explanation	Note
D	Block is encrypted	-
V	Block is verified	-
X	Block is tampered	-
T	Block has transmitted	Offline status
U	Block hash non-transmitted	Transmitted when it is online

- Time and date of block creation

Time and date of block creation is recorded every second when the previous block is connected to the current block to complete the block, and the block header is registered on the block management file.

4. Evaluation

Three scenarios of blockchain encoding, tampering verification, and identification were carried out for algorithm performance analysis. The verified information for each scenario is explained in Table 6.

Table 6. Performance analysis scenario.

Type	Information
Blockchain encoding	30 actual operation data used to check the light blockchain encoder program operation
Tampering verification	Tampering status verified for 30 files with blockchain encryption
Tampering identification	1byte data modification made to 15 files, followed by verification of tampering status

- DTG device operation record data file (Operation record for 30 days)

A set of tests was prepared to check the operation status of the light blockchain program, as shown in Table 7.

Table 7. Digital Tachograph data file.

No.	File Size (Byte)	No.	File Size (Byte)	No.	File Size (Byte)	No.	File Size (Byte)	No.	File Size (Byte)
1	319,877	7	9,316,777	13	5,421,229	19	4,669,632	25	3,697,487
2	1,105,437	8	7,537,644	14	5,400,939	20	4,558,907	26	3,346,254
3	1,172,124	9	6,840,120	15	5,209,989	21	4,540,726	27	3,238,979
4	482,701	10	6,554,882	16	5,037,681	22	4,289,195	28	3,038,636
5	1,950,477	11	6,516,887	17	4,845,035	23	4,067,517	29	2,524,024
6	1,070,084	12	6,213,691	18	4,703,952	24	3,890,020	30	2,047,243

- .blocklist.dat (Initial state)

.blocklist.dat is the header that internally manages the light blockchain. It manages the list of light blockchain headers and records the internal blockchain status of the device, such as data tampering status and block hash transmission status. In the initial status of the DTG device, the Genesis Block information is recorded, as shown in Table 8 below. The block number is 00000, which is automatically generated upon registering the device. The user manages the light blockchain, which is generated on the user domain upon user registration. Multiple users can be registered on the device.

Table 8. Blockchain header file samples in normal status.

Block No.	Date	File Name	File Size	Prev Hash	Block Hash	Status	Time Date	Note
00000	20191110	0000020191110	000000000	a48c1ec941 2e5a66df09 503b3cdcb4 b8a16d085	43f3e0a4aa 388e1eef51 9f55118f5b e3400dd989	D	20191111 04:49:44	Genesisblock
00001	20191111	0000120191111	000319877	43f3e0a4aa 388e1eef51 9f55118f5b e3400dd989	9a51aa0804 dacf2eb064 ad8edfd8f3 b342441db5	V	20191211 18:06:00	Block1
00030	20191210	0003020191210	002047243	3a624997e5 695044c8bb 7f1ac9023b 71493c3fff	fda221a900 5269fc852d c4aa340117 ff0687a90a	V	20191211 18:06:00	Block30

- blockchain header file (.blocklist.txt)

The light blockchain header list file is recorded in the order of the block number and protects data from tampering after block encryption is performed. The data on the tachograph is recorded in units of one day. Therefore, after data in one-day units is created, the list encrypted through an encoding process manages the tampering status of blockchain files. This algorithm performs the encoding process in units of blocks. As shown in Table 8, only the Genesis Block is flagged as 'D', and all the other blockchain headers have been changed to 'V', showing that the data file in the current system has been verified.

- Results of tampering detection

1-byte letters were changed in 15 data files among 30 files before performing the tampering detection. The results of decoding are flagged on the block status part of the blocklist.txt file. If any tampering is detected after decoding, the block status value is flagged as 'X', and if there is no tampering, it is flagged as 'V'. Table 9 shows the detection results.

Table 9. Blockchain header file samples in tampered status.

Block No.	Date	File Name	File Size	Prev Hash	Block Hash	Status	Time Date	Note
00000	20191110	0000020191110	000000000	a48c1ec941 2e5a66df09 503b3cdcb4 b8a16d085	43f3e0a4aa 388e1eef51 9f55118f5b e3400dd989	D	20191111 04:49:44	Genesisblock
00001	20191111	0000120191111	000319877	43f3e0a4aa 388e1eef51 9f55118f5b e3400dd989	9a51aa0804 dacf2eb064 ad8edfd8f3b 342441db5	X	20191211 18:20:24	Block1
00030	20191210	0003020191210	002047243	3a624997e5 695044c8bb 7f1ac9023b 71493c3fff	fda221a900 5269fc852d c4aa340117 ff0687a90a	V	20191211 18:20:24	Block30

- Results analysis of program execution

The operation sequence of the tamper-resistant algorithm for digital tachographs is first, the data creation process, second, the data encoding process, and third, the tamper detection process. In this research, 30 days of operation records were encoded in blockchains, and the encoded data was changed arbitrarily to test for the file's tamper detection. As a result, all 15 modified files were detected accurately, proving the usability of the program.

The system used to measure the execution time of the light blockchain algorithm was NVIDIA AGX Xavier (NVIDIA, Santa Clara, CA, USA) which has 8Core ARM V8.2 64 bit CPU (Clock speed 1.2–2.265 GHz) and 512Core VoltaGPU, 32 GB LPDDR4x memory. The OS is Ubuntu 18.04LST, and the algorithm is implemented by using Go language version 13.0 and only the ARM CPU.

Table 10 shows the descriptive statistics analysis of the data measured values for algorithm execution time, and the time consumption for the encoding and decoding of each file is shown in msec/file. The Mbyte/file unit shows the size of the operation record data file in megabytes. The size of the operation record data file differs according to the operation time and distance, so it is not meaningful to compare the units of the files as they are. To compensate for this, the program execution time per megabyte is displayed to enable the relative comparison using msec/Mbyte units considering the file capacity in encoding and decoding time calculation.

Table 10. Encoding & Decoding Time Descript Statistic Result.

	Encoding		Decoding	
	msec/File	msec/Mbyte	msec/File	msec/Mbyte
Mean	5.636	1.856	5.575	1.648
Standard Deviation	2.348	1.778	2.340	0.874
Minimum	1.482	1.216	1.579	1.214
Maximum	12.721	10.351	11.755	5.524
Sum	163.449	53.824	161.668	47.801
Count	29	29	29	29

An analysis of the descriptive statistics shows that the encoding processing performance for Mbyte units is 1.85 ms/Mb on average, and the decoding time is 1.64 ms/Mb. Therefore, for the 29 files with 118.7 Mbyte, the encoding time is 219.6 ms, and the decoding time is 194.6 ms. For the data during six months, the encoding time is estimated as 1.319 s and the decoding time as 1.16 s. The execution time per file is 5.6 ms for encoding on

average and 163.4 ms for encoding all 29 files. Furthermore, the decoding time is 5.5 ms on average, and the execution time for all 29 files is 161.6 ms.

The parameter statistics analysis for program execution time requires the preprocessing of data. Methods for data preprocessing include refining the missing value and outliers by inserting the average, creating, and processing data using distribution function, controlling the included line, etc. Outliers can be identified intuitively by using a boxplot function. Figure 8 shows the data that is deviated from the confidence interval of 99.3% in circles. 99.3% means range of boxplot. Data outside this range influences the average calculation excessively and may even influence the normality of the data. In this research, the outliers were removed by deleting the data lines [35].

Figure 8. Boxplot with outlier (R Studio 1.3).

After removing the outliers, check the box graph to see that the outliers have been removed, as shown in Figure 9. This shows a boxplot using 19 data, excluding outliers from all 29 data. The average encoding and decoding time per file are 6.15 ms and 6.01 ms, and the average encoding and decoding time per Mbyte is 1.32 ms and 1.29 ms.

Figure 9. Boxplot without outlier (R Studio 1.3).

The T-test is performed to analyze program execution time due to the issues in the population mean estimation. The data need to be reviewed as to whether it has a normal distribution to perform the T-test. The typical methods for evaluating normality are the Shapiro-Wilk normality test and the Kolmogorov-Smirnov test. When the sample size is 2000 or less, the Shapiro-Wilk test can be performed for verification. The null hypothesis and alternative hypothesis for the Shapiro-Wilk normality test is as follows:

$$\begin{cases} H_0 : \text{The population of the sample is in normal distribution.} \\ H_1 : \text{The population of the sample is not in normal distribution.} \end{cases} \quad (2)$$

An analysis of the results shows that the p -value is greater than 0.05, and the null hypothesis cannot be rejected, as shown in Table 11. Therefore, the encoding and decoding time in file units and the encoding and decoding time in Mbyte units can both be said to have a normal distribution.

Table 11. Shapiro-Wilk Normality test (R-Studio 1.3).

Type	Normality Examination Results	Interpretation
Encoding/file	$W = 0.9637, p\text{-value} = 0.6201$	Normal distribution
Encoding/Mbyte	$W = 0.93169, p\text{-value} = 0.1664$	Normal distribution
Decoding/file	$W = 0.94886, p\text{-value} = 0.3501$	Normal distribution
Decoding/Mbyte	$W = 0.93761, p\text{-value} = 0.216$	Normal distribution

A normal distribution examination can be performed after removing missing values or outliers as data processing, as shown in Table 11. If it qualifies the normal distribution examination in a 0.05 significance level, a T-test can be performed. The statistics program used for T-test analysis was R 4.03 Windows10 64 version. The population mean can be estimated through One sample T-test function. The null hypothesis and alternative hypothesis used for the population mean estimation is as follows:

$$\begin{cases} H_0 : \mu = 0 \\ H_1 : \mu \neq 0 \end{cases} \quad (3)$$

Table 12 shows the confidence intervals of processing time per Mbyte and processing time per file as performance indicators of the DTG device. The estimation results of the population mean were tested at a 0.05 significance level. In this case, the performance index per Mbyte is more useful for system design. The p -values of the encoding and decoding time per Mbyte are equal to 2.2×10^{-16} and this is below the significance level, thus rejecting the null hypothesis and accepting the alternative hypothesis. The estimated mean value of encoding time is 1.324 ms and the 95% confidence interval is <1.285 ms, 1.364 ms>. Additionally, the estimated mean value of decoding time is 1.290 ms and the 95% confidence interval is <1.264 ms, 1.315 ms>.

Table 12. One-sample T-test (R-Studio 1.3).

	Encoding		Decoding	
	msec/File	msec/Mbyte	msec/File	msec/Mbyte
T-value	23.259	70.82	21.776	105.6
df	19	19	19	19
p -value	2.014×10^{-15}	2.2×10^{-16}	6.748×10^{-15}	2.2×10^{-16}
95% interval min	5.599	1.285	5.429	1.264
95% interval max	6.706	1.364	6.584	1.315
Mean	6.153	1.324	6.007	1.290

Table 13 analyzes the difference in encoding time estimated using average execution time according to t-test, with the actual time. This table shows the utility of estimated value in file units with the average estimated value of execution time in Mbyte capacity. The estimation error of the encoding time in Mbyte units is -1.5% , and the decoding time is -3% . The estimation error of execution time in file units is 9.1% and 7.74% , showing that the execution time can be predicted accurately by estimating the execution time in Mbyte units.

Table 13. Calculation of estimation error.

	Encoding/File	Encoding/Mbyte	Decoding/File	Decoding/Mbyte	Note
Average time estimated value	6.153	1.324	6.007	1.290	T-test results
Estimated file processing time	178.422	160.995	174.191	156.764	Processing time for 29 files calculated by average estimated value
Encoding estimation error (ms)	14.974	-2.454	-	-	Error = Estimated value – execution time
Decoding estimation error (ms)	-	-	12.523	-4.904	Error = Estimated value – execution time
Estimation error rate %	9.161	-1.501	7.746	-3.033	Error rate

5. Conclusions

With the increased social attention towards driving safety, many devices and systems are being introduced for the purposes of preventing accidents. Since speeding and overloading commercial vehicles are the leading causes of large-scale traffic accidents, it is the international trend to mandate vehicles to install tachographs, submit operation data regularly, or inspect and demand the submission of operation data for traffic control. However, there are many routes available to tamper with operation records in DTG devices. Preventing this is critical for improving the reliability of DTG devices and for the settlement of the systems.

This research suggests a tamper-resistant light blockchain algorithm that operates smoothly, even in low level hardware of the digital tachographs and a mechanism where light blockchain works in unstable communication situations or in situations where communication is disabled due to various reasons. The proposed light blockchain algorithm took an average time of 1.85 ms/Mb for encoding and 1.65 ms/Mb for decoding. When outliers were removed by a statistical test, the average encoding time was estimated as 1.32 ms/Mb , and the average decoding time was estimated as 1.29 ms/Mb . All of the tampered data were detected in the tampering detection tests.

According to the statistical estimation method, the average execution time was estimated as a performance index of light blockchain software. The estimation error in the execution time results in file units turned out to be about three times greater. Therefore, the statistical analysis confirmed that it is effective to design and apply performance indicators of the system using the estimated execution time in Mbyte.

The proposed algorithm is devised to be suitable for low-level hardware and be applied to all types of mobile devices and various IoT (Internet of Things) fields that need to be operated enclosed.

Author Contributions: Conceptualization, Y.K. and J.K.; methodology, J.K.; software, Y.K.; validation, J.B., Y.K. and J.K.; data curation, Y.K. and J.K.; writing—original draft preparation, Y.K.; writing—review and editing, J.K.; supervision, J.K.; project administration, J.K.; funding acquisition, J.B. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was granted by Ministry of Land, Infrastructure and Transport of Korea, 20CTAP-C152294-02.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Korea Transportation Safety Authority and are available at <http://www.kotsa.or.kr/eng/main.do> (accessed on 2 March 2021) with the permission of Korea Transportation Safety Authority.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. OECD Data, Road Accidents. Available online: <https://data.oecd.org/transport/road-accidents.htm> (accessed on 9 February 2021).
2. Joint Research Center Digital Tachograph. Available online: <https://dtc.jrc.ec.europa.eu/> (accessed on 3 December 2020).
3. ELD Mandatory for Land Trucks in US, Increase of Distribution Costs. Available online: http://www.centralx.co.kr/bbs/board.php?bo_table=notice&wr_id=21&sfl=&stx=&sst=wr_hit&sod=asc&sop=and&page=1 (accessed on 1 December 2020).
4. Management Guidelines for Automobile Operation Records and Devices. Korea Ministry of Land, Infrastructure and Transport Announcement No. 2020-273. Available online: <http://www.molit.go.kr/USR/policyTarget/dtl.jsp?idx=311> (accessed on 1 December 2020).
5. Traffic News. Voices Raised to Reexamine DTG Operation Record Transmission System. Available online: http://www.bstci.or.kr/xe/index.php?mid=safte_info_board&document_srl=62790 (accessed on 27 July 2020).
6. Baek, S.; Sohn, M. Design and implementation of stable data storage for digital tachographs. *Korea Inf. Process. Soc. J.* **2012**, *1*, 71–78. [CrossRef]
7. Park, G. Design and Implementation of TPM-Based Tamper-Resistant Digital Tachographs. Available online: <http://syscore.sejong.ac.kr/~{}woongbak/publications/DJ12.pdf> (accessed on 31 October 2020).
8. Kim, G. Korea Patent No.1020954340000. Korean Intellectual Property Office. Available online: <https://doi.org/10.8080/1020180109087> (accessed on 22 July 2020).
9. ATEC T & Co., Ltd. Intelligent Driver Assistance System Equipped with Accident Video Transmission Function. Advanced Driver Assistance System with Accident Video Transmission. Korea Patent No. 1020180138027, 12 November 2018. [CrossRef]
10. Nakamoto, S. Bitcoin P2P e-Cash Paper. Available online: <https://archive.is/DUcfX> (accessed on 31 October 2020).
11. Litecoin Foundation. 2020. Available online: <https://litecoin-foundation.org/about-us/> (accessed on 1 December 2020).
12. Ethereum Foundation. Available online: <https://ethereum.org/en/foundation/> (accessed on 5 February 2020).
13. Ripple Company. Available online: <https://ripple.com/company/> (accessed on 5 December 2020).
14. Bitcoincash.Org. Available online: <https://www.bitcoincash.org/> (accessed on 5 December 2020).
15. EOS. Available online: <https://eos.io/> (accessed on 4 December 2020).
16. Poon, J.; Dryja, T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Available online: <https://lightning.network/lightning-network-paper.pdf> (accessed on 5 December 2020).
17. Poon, J.; Buterin, V. Plasma: Scalable Autonomous Smart Contracts. Available online: <https://plasma.io/plasma.pdf> (accessed on 5 December 2020).
18. Hong, J. Side Chain the Solution to Blockchain Scalability. Available online: <https://medium.com/@joycehong0524/blockchain-scalability-solution-side-chain-feat-plasma-f3a530cd482c> (accessed on 5 December 2020).
19. Dai, P.; Mahi, N.; Earls, J.; Norta, A. Smart-Contract Value-Transfer Protocols on a Distributed Mobile Application Platform. Qtum Whitepaper, Qtum Foundation. Available online: https://qtum.org/user/pages/01.home/Qtum%20New%20Whitepaper_en.pdf (accessed on 5 December 2020).
20. Nem Group. Available online: <https://nem.io/about> (accessed on 5 December 2020).
21. Hdac Technology AG. Available online: <https://www.hdactech.com/en/company/about.do> (accessed on 5 February 2021).
22. Liu, Y.; Wang, K.; Lin, Y.; Xu, W. LightChain: A Lightweight Blockchain System for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3571–3581. [CrossRef]
23. Bai, L.; Hu, M.; Liu, M.; Wang, J. BP1IoT: A Light-Weighted Blockchain-Based Platform for Industrial IoT. *IEEE Access* **2009**, *7*, 58381–58393. [CrossRef]
24. Gruber, D.; Li, W.; Karame, G. *Unifying Lightweight Blockchain Client Implementations*; Workshop on Decentralized IoT Security and Standards (DISS): San Diego, CA, USA, 2018.
25. Lin, T.; Yang, X.; Wang, T.; Peng, T.; Xu, F.; Lao, S.; Ma, S.; Wang, H.; Hao, W. Implementation of High-Performance Blockchain Network Based on Cross-Chain Technology for IoT Applications. *Sensor* **2020**, *11*, 3268. [CrossRef] [PubMed]
26. Yue, L.; Junqin, H.; Shengzhi, Q.; Ruijin, W. Big data model of security sharing based on blockchain. In Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), Chengdu, China, 10–11 August 2017; pp. 117–121.

27. Xiong, Z.; Zhang, Y.; Niyato, D.; Wang, P.; Han, Z. When mobile blockchain meets edge computing. *IEEE Commun. Mag.* **2018**, *56*, 33–39. [[CrossRef](#)]
28. Yuan, Y.; Wang, F. Towards blockchain-based intelligent transportation systems. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 2663–2668.
29. Kang, J.; Yu, R.; Huang, X.; Wu, M.; Maharjan, S.; Xie, S.; Zhang, Y. Blockchain for Secure and Efficient Data Sharing in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2019**, *6*, 4660–4670. [[CrossRef](#)]
30. Didouh, A.; Lopez, A.B.; Hillali, Y.E.; Rivenq, A.; Faruque, M.A.A. Eve, You Shall Not Get Access! A Cyber-Physical Blockchain Architecture for Electronic Toll Collection Security. In Proceedings of the 2020 IEEE 23th International Conference on Intelligent Transportation Systems (ITSC), Rhodes, Greece, 20–23 September 2020; pp. 1–7. [[CrossRef](#)]
31. Lee, S.; Jung, H.; Han, B. Security Assured Vehicle Data Collection Platform by Blockchain: Service Provider’s Perspective. In Proceedings of the 2019 21th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Gangwon-Do, Korea, 17–20 February 2019; pp. 265–268. [[CrossRef](#)]
32. Bera, B.; Saha, S.; Das, A.K.; Kumar, N.; Lorenz, P.; Alazab, M. Blockchain-Envisioned Secure Data Delivery and Collection Scheme for 5G-Based IoT-Enabled Internet of Drones Environment. *Proc. IEEE Trans. Veh. Technol.* **2020**, *69*, 9097–9111. [[CrossRef](#)]
33. Saldamli, G.; Karunakaran, K.; Vijaykumar, V.K.; Pan, W.; Puttarevaiah, S.; Ertaul, L. Securing Car Data and Analytics using Blockchain. In Proceedings of the IEEE 2020 Seventh International Conference on Software Defined Systems (SDS), Paris, France, 20–23 April 2020; pp. 153–159. [[CrossRef](#)]
34. Lei, A.; Cruickshank, H.; Cao, Y.; Asuquo, P.; Ogah, C.P.A.; Sun, Z. Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems. *IEEE Internet Things J.* **2017**, *4*, 1832–1843. [[CrossRef](#)]
35. Oracle. Available online: https://docs.oracle.com/cd/E17236_01/epm.1112/cb_statistical/frameset.htm?ch07s02s10s01.html (accessed on 2 December 2020).