



# A decentralized vehicle anti-theft system using Blockchain and smart contracts

Debashis Das<sup>1</sup> · Sourav Banerjee<sup>2</sup> · Uttam Ghosh<sup>3</sup> · Utpal Biswas<sup>1</sup> · Ali Kashif Bashir<sup>4</sup>

Received: 23 October 2020 / Accepted: 9 February 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

## Abstract

In recent years, vehicle theft has been increasing remarkably. It is a stigma to our society. The impacts of vehicle theft have been drastically affecting the social safety and economic condition of the whole world because of unavailability of a proper theft detection mechanism. The few existing vehicle anti-theft systems suffer from major problems such as the leakage of personal information, centralized-based system, proper key management, and data security. In this paper, a decentralized Blockchain-based Vehicle Anti-Theft System (BVATS) is proposed to overcome these problems using smart contracts. Blockchain is a very cutting-edge decentralized technology that is well-equipped with data immutability and a secure information-sharing platform. The smart contract is a digital agreement, which can authenticate an entity automatically and stores information by verifying the predefined condition(s). This paper also explains how Blockchain can be adopted for vehicle security and provides a stepwise implementation of the proposed methodology by providing test bed results and significant critical comparison analysis with other existing systems. Using the BVATS, more than one person can be authorized to drive a vehicle without hampering the vehicle data and maintaining security.

**Keywords** Blockchain · Smart contract · Vehicle anti-theft system · Vehicle security management · Decentralized vehicle anti-theft system · Blockchain for vehicle security

## 1 Introduction

Vehicles have a wide range of implications in social interactions. They provide a prevalent and reliable mode of transportation in our society. Nevertheless, in many developed countries, vehicle theft is a concern for societies. Anti-theft systems ensure protection of vehicles from thieves. The anti-theft system [1] is a method to prevent unauthorized access to any valuable items. For every stolen vehicle, the owner needs to compensate the

loan premiums legitimately, though the stolen vehicle is going to return. If the number of robberies decreased, the insurance rates might decrease as well. A cyber-attack [2] can steal and 1 manipulate a large amount of confidential data [3]. Many scammed vehicles are also used for other criminal acts, so the impact on the community is also quite substantial [4]. The Government department, police authorities, and other agencies are working closely with industries by sharing

This article is part of the Topical Collection: *Special Issue on Blockchain for Peer-to-Peer Computing*

Guest Editors: Keping Yu, Chunming Rong, Yang Cao, and Wenjuan Li

✉ Ali Kashif Bashir  
a.bashir@mmu.ac.uk

Debashis Das  
debashis2124@gmail.com

Sourav Banerjee  
mr.sourav.banerjee@ieee.org

Uttam Ghosh  
uttam.ghosh@vanderbilt.edu

Utpal Biswas  
utpal0172@gmail.com

<sup>1</sup> Department of Comp. Sc. and Engg. University of Kalyani, Kalyani, India

<sup>2</sup> Department of Comp. Sc. and Engg. Kalyani Govt. Engg. College, Kalyani, India

<sup>3</sup> Vanderbilt University, Nashville, TN, USA

<sup>4</sup> Manchester Metropolitan University, Manchester, UK

intelligence to fight against vehicle theft issue. They are investing billions of dollars for getting efficient security features such as tracking devices and key encryption strategies [5, 6] to tackle vehicle-stealing issue, vehicle thefts can be reduced.

As the Blockchain is a low-cost technology and a useful thing, we can utilize this for the vehicle with an efficient and secure anti-theft solution. Blockchain can provide trust and secure services in many different areas such as healthcare management [7], supply chain management [8], smart appliances [9], asset management [10], cross-border payment [11], and so on. The smart contract is a digital agreement, which can be used to authenticate vehicles' user automatically and store information of vehicles by checking the predefined condition(s). Thus, we have designed a decentralized vehicle anti-theft system using Blockchain and smart contract to provide the security of vehicles.

This research work presents an efficient and secure Blockchain-based Vehicle Anti-Theft System (BVATS) architecture. In this architecture, every vehicle owner needs to connect in the blockchain network to access the services, which are available in the BVATS. Vehicle-related organizations, mentioned in section 3, can observe a vehicle status and activities of the vehicle's owner. The owner can start the vehicle after verifying the owner data by a biometric device [12]. The verification data can be replaced by the deployed smart contract automatically using the wireless Internet of Things (IoT) [13]. A Control Area Network (CAN bus) [14] is a powerful vehicle bus model that can be built to connect microcontrollers and accessories in a device without a hosted machine. It is a message-based protocol, basically designed for multiplex electrical wires in vehicles. In particular, it is a framework that transmits data through an On-board Diagnostics (OBD-II) connector using a wireless Internet interface framework. OBD-II connector is also an on-board computer that can monitor emissions, mileage, speed, about a vehicle. Its connected to the check engine light. It is essential, simple, and particularly suitable for the access and transmission of the data. So, we can get speed of a vehicle using OBD-II connector.

The BVATS has been proposed using the blockchain technology, which intensifies to protect vehicle theft issue. Only a vehicle owner can drive his vehicle using centralized based existing systems. The significant improvement in the BVATS is that anyone can drive the vehicle authorizing by the vehicle owner. The BVATS also provides an adequate key management artifice and gives assurance from the stealing of vehicles.

The main contributions of this research work are given in the following:

- 1) This research work presents the BVATS architecture to protect vehicles from unauthorized access by developing a smart contract.
- 2) An unauthorized access detection algorithm is presented and implemented in this paper.
- 3) An implementation method and a set of testing results have been provided concisely to understand the working process of the BVATS.
- 4) It is shown that more than one driver's data can be added and removed using the smart contract by an authenticated vehicle owner.

The rest of the paper organizes in the following manner. Section 2 provides a brief literature survey of existing anti-theft vehicle systems and their limitations. Section 3 describes the system architecture and implementation methods of the proposed architecture. Also, the proposed algorithm for unauthorized access detection is demonstrated in section 3. The tested scenarios of the developed smart contracts for the proposed architecture are shown in section 4. Therefore, section 5 describes the performance analysis of the BVATS and shows the key features and benefits of BVATS over the existing vehicle anti-theft methods. Finally, the conclusion and future directions are summarized in section 6.

## 1.1 Vehicle security issues

In the absence of the owner, vehicle can be thefted in a parking area or any other insecure place. Applications of the Internet of Things (IoT) in vehicle routing and communication raises several types of security and safety issues such as Internet of Vehicles (IoV) threats, Denial of Service Attack, False Message Injection, and Malware Attack [15, 16]. Therefore, an adequate solution is required to solve these types of issues in the IoT environment. The existing methods [12, 17–19] also have some potential risks for vehicle safety from cybercriminals as many communication systems are connected with the Internet. Security failure of these methods may contribute to the leakage of personal information, a security vulnerability in the mobile application, and system manipulation. Without addressing these issues, user privacy and data security could be suffered. Because of the increasingly exposed working situation of embedded systems, the system becomes malfunctioned with potentially unprotected threats. Attackers reprogram a stolen embedded system and continue to use it [20]. The area is concerned with the normal protection countermeasures to avoid access to the network by information validation, cryptographic strategies for protecting data integrity, and network defense mechanisms. There are many kinds of threats that originated at taking advantage of mobile computing growth. Now, two variants of algorithms known as A5/1 and A5/2 have been implemented in the Global System

for Mobile Communication (GSM) network. As the encryption algorithm has been released, it has been shown that encryption can be cracked within approximately 6 h [21]. The IoT system has insufficient management and technology for security. Since this functionality, there are several risks in the IoT environment as it is a low specification system, where security cannot be applied easily [22].

## 1.2 Blockchain in vehicle security

Blockchain [23] is a decentralized architecture, which allows users to form a collective agreement without involving middleman or intermediaries such as a government or a company. Data security of blockchain can provide confidentiality for all the sensitive data stored on it. Blockchain can store the tracking information of communication between vehicles [24]. It can be implemented for vehicle registration and traffic violators tracking [25]. Also, it can be applied to provide an efficient and secure antitheft system for vehicles. Blockchain, connected through the peer-to-peer (P2P) network [26]. Blockchain Ledger can store data with the amenity of data immutability [27].

## 1.3 Smart contracts in vehicle security

Contract regulation is the main factor of the economy, which involves contracts of all economic transactions between two or more groups, along with the government agencies. Thus, Smart contract [28] can be used to provide vehicle protection through secure authentication and verification. It acts as a usual contract but performs digitally and automatically. It can be transformed into code and run on a blockchain network. It is an application to execute an automatic event based on a common agreement [29, 30]. The smart contract can be destroyed and replaced by a new one using the SELFDESTRUCT function [31–34]. Smart contracts are digital agreements, which can be applied for many use cases such as finance, e-government, IoT, energy management, and security management [35]. There are three steps to perform a smart contract [36]. At the first level, the agreement is formed in code for some participants and deployed on a blockchain platform. Secondly, the smart contract is called by an event and ready to execute itself. Finally, the smart contract execution is completed while the set of conditions is met and then triggered some events for further processing. The smart contract is deployed to a Virtual Machine such as Ethereum Virtual Machine (EVM) [37] that can store data [38]. We can use two types of variables in solidity: state variables and local variables. These variables can store the state of the smart contract by storing the values in a block on the

blockchain. State variables are storage by default, where information is stored in the blockchain [39].

## 2 Literature review

Since the year 1990, many researchers have given their anti-theft solutions for vehicles. But, vehicle theft was not prevented completely.

Mrimoy et al. [17] proposed an anti-theft system for the protection of vehicles using GSM and Global positioning System (GPS) in the year 2017. They used a fingerprint enabled verification system to authenticate the driver. A disadvantage of this system is that no one can drive the vehicle without the vehicle owner. Wei et al. [40] designed a prototype to locate vehicles with enabling the tracking system using the GSM Network and the BeiDou Navigation Satellite System (BDS). In 2015, Shruthi et al. [19] developed an anti-theft system with a smartphone application. With a mobile application, this system can trace the stolen vehicle. They focused on the simple and cost-effective vehicle monitoring device utilizing GPS and GSM technology as well as a Smartphone framework for the detection of any moveable resources.

In 2013, Zhigang et al. [18] presented an anti-theft tracking system with the help of IoT. The use of IoT leaves their system vulnerable to cyber-attacks. Attackers can mangle confidential information. Dashora et al. [14] proposed an IoT enabled architecture to detect the vehicle accident. They developed a device that will provide urgent support from the hospital once the vehicle meets the accident, and the driver is unable to contact the ambulance. Prevalence problems like positioning confidentiality and application safety involve other specific IoT circumstances such as positioning. In a term, IoT devices work with fewer assets and network defenses in further unprotected conditions. We need to develop lightweight strategies to challenge this hazardous area. Mbarek et al. [41] developed a lightweight system for IoT applications. They focused on reducing blockchain processing and communication costs using mobile agents.

Wei et al. [40] suggested a cost-effective distributed vehicle wireless locating and monitoring model framework. This framework is comprised of just two components: a smartphone app terminal for a user and a vehicle terminal to capture vehicle positions in real-time. Communication-related threats happen in mobile network system architecture and management malfunctions. The hacker may attempt to mangle mobile network encryption [20]. In 2018, Qian et al. [42] designed a theft tracking and alarming system for vehicles using an android-based operating system. When utilizing a device's built-in devices, such as a smartphone, the vehicle's suspicious details can be transmitted through a communication network to the mobile phone of its owners.

Liu et al. [43] proposed a low-cost vehicle anti-theft system using the idea of PhoneInside by leveraging an extinct smartphone. In the proposed system, the smartphone can detect vehicle location during driving, vehicle activity, and track GPS based location. The identification of vehicle theft can be detected using ad hoc authentication. A timely alarming system and message tracking for fast salvation can be performed using this system.

Liu and He [44] proposed an anti-theft alert system for vehicles applying face identification. This system can catch an unauthorized person because of a facial recognition mismatch and send an image of the unauthorized person to the police station or the vehicle owner. Once the vehicle starts moving, this device captures the video pictures of the driver using an activated IR illumination tool and then detects the eyes of the driver and identifies the face of the driver applying the PCA algorithm. However, the vehicle tracing method has not been included in this system.

Yu et al. [30] proposed a distributed energy transaction mechanism using blockchain smart contracts, which includes auditing, bidding, and settlement. Mohanta et al. [27] described the distinct components and working principle of smart contracts. They also remarked and analyzed the different use cases of smart contracts. Alotaibi [23] analyzed recent security advancements in blockchain to resolve IoT constraints. They showed that blockchain endeavors to address IoT cybersecurity vulnerabilities that are listed in four categories: end-to-end traceability, transparency and security of information, identity verifications, and authentications, and confidentiality, data integrity, and availability (CIA).

### 3 Blockchain-based vehicle anti-theft system (BVATS)

This section describes the system architecture and implementation method of the proposed Blockchain-based Vehicle Anti-Theft System (BVATS), where a smart contract has been developed to verify a vehicle's driver. A driver can be a vehicle owner or an authorized person selected by the vehicle's owner. Only the push-start vehicles equipped with the keyless entry system has been considered in the BVATS architecture. Because the automobile industry is getting advanced at a rapid scale. Most of the advanced vehicles are offering keyless features. Moreover, if someone has a key used vehicle, it is not required to verify using the biometric verification at the time of ignition. The doors can be opened by any means. Whether, a biometric system is required to verify the driver, thus only keyless vehicles are considered in this system.

### 3.1 An overview of the BVATS architecture

The proposed BVATS can protect the vehicle from stealing by detecting unauthorized access to it. Six nodes are taken in the BVATS architecture. These nodes are the Owner, Vehicle Seller Agency (VSA), Vehicle Certification Agency (VCA), Blockchain Server, Vehicle Transportation Agency (VTA), and the Vehicle. More number of nodes can be taken if necessary. In the BVATS architecture, two attributes are used such as Universal Vehicle Key (UVK) and Authorized Driver (AD).

The responsibility of each node and attribute in this proposed architecture are described in detail in the following:

**UVK:** A Vehicle Engine Number (VEN) can identify a vehicle. An Owner's License Number (OLN) can recognize a vehicle's owner. But, the UVK can recognize the vehicle as well as the vehicle's owner. The UVK is a random sequence that is formed using the owner's license number and the vehicle engine number. The process of the UVK generation is automated. The VCA, an authentication entity which can only insert the VEN and OLN as input to the smart contract. The smart contract then automatically generates the UVK by combining these two numbers with a random sequence in such a way that no one can assume the actual UVK as well as the policy of the key generation process. The terms and conditions of the policy can be decided by the administrative authorities. This key can be accessed by the smart contract internally to authenticate an owner and to identify a vehicle.

**Owner:** In the proposed BVATS architecture, the owner is a vehicle buyer. While buying a vehicle, the owner needs to provide necessary documents for registration to the VSA. The owner is a connected node and does not need to store a copy of the whole blockchain ledger. The owner can store their transaction records whenever it is required.

**AD:** AD is the authorized driver who can be an owner's friend or personal driver. Whenever they want to drive the owner's vehicle, the owner has to add their information (i.e., driving license, biometric data, etc.) to the blockchain ledger through using the smart contract on the application site.

**VSA:** Every person must need to buy a vehicle from a VSA. Every purchaser must need to provide their genuine credentials to a VSA. The VSA collects and authenticates the purchaser's information. The VSA prepares a set of compound data combining the information of the owner (i.e., Government ID proof, Mobile Number, etc.) and purchased vehicle (i.e., chassis number, make, and model, etc.). Finally, the VSA forwards those data to the VCA.



**VCA:** The VCA generates the UVK using the owner's information and matches the information with the owner. It keeps a blockchain ledger to observe owner activities if needed. Here, it is one of the minor's nodes in the network. It is also responsible for sending the owner's login details to the owner.

**Blockchain Ledger:** Blockchain Ledger is the decentralized computer(s) where all the blockchain nodes are owners and organizations. Smart contracts are written in code and deployed on the blockchain platform. Then an event (i.e., owner authentication, driver authorization) will get triggered by the execution of the smart contract that is called by the vehicle owner. When this happens, the smart contract is executed. If the smart contract execution is completed, it can store the information. The blockchain ledger can maintain participant's identities, their cryptocurrency balances. The executed transactions can be recorded on the ledger. Thus, Blockchain ledger is more important to access the vehicle data with greater security in this proposed architecture.

**VTA:** The VTA is responsible for tracking the vehicle when unauthorized access occurs. For the unauthorized access, the vehicle's device notifies to the vehicle owner and to the VTA. When the owner is away from the vehicle, then the VTA will be tracing the vehicle until receiving the vehicle status from the owner. The status contains a safe or unsafe state of the vehicle. The VTA also keeps a copy of the blockchain ledger to identify and trace the vehicle to take immediate action if needed.

**Vehicle:** Push-start enabled a biometric device is installed to the vehicle for authorizing the driver. The vehicle's storage device is equipped to store and receive information. The information contains the vehicle velocity, the current driver (who drives the vehicle now), and biometric data of the driver. Besides, the driver data can be synchronized with the vehicle device using the wireless Internet of Things (IoT) technology.

The proposed BVATS architecture is shown in Fig. 1, which illustrates the communication process between the present nodes in the BVATS architecture in the following steps:

**Step (1):** Every vehicle owner must have to provide their credentials to the VSA, which is responsible for verifying and collecting vehicle owner information physically. An owner gives the necessary information to the VSA for registration while buying a vehicle. For this, the owner needs to contact the nearest or preferable VSA and can be registered for the purchased vehicle. The VSA can verify owner information at the time of vehicle registration. It can authenticate the owner's credentials with the owner directly (head to head verification), where verification of the originality of the owner's information can be maintained. So, no other person can buy a vehicle by using the stolen credentials from somewhere. Thus, VSA is required to verify the originality of a vehicle owner credentials in this proposed architecture.

**Step (2):** The VSA forwards the owner's credentials along with vehicle information to the VCA. Then the VCA generates a UVK for the owner.

**Step (3):** The VCA stores the owner's credentials and the UVK to the blockchain database. The VCA sends an interface link to the owner for accessing the blockchain accounts.

**Step (4):** The owner needs to authenticate in the blockchain network to call a smart contract. He can call a smart contract after authenticating to the network, where the smart contract is deployed to the blockchain by VCA.

**Step (5):** The owner needs to authorize through the biometric device for driving the vehicle. When the owner permits another person for driving the vehicle, the owner needs to add that person's authorization data to the

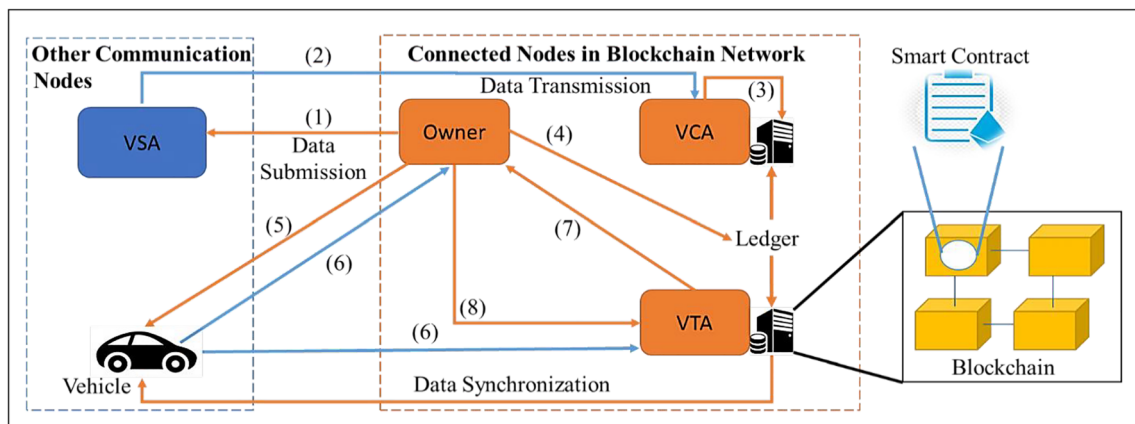


Fig. 1 System overview of the BVATS architecture

blockchain server. After a driving session, added data will be removed by the owner.

**Step (6):** If an unauthorized person tries to access the vehicle without the owner's permission, then a notification will be sent to the owner and to the VTA after checking the vehicle speed.

**Step (7):** Then the VTA traces the vehicle continuously and asks for vehicle status, which will be provided by the vehicle owner.

**Step (8):** Finally, The owner updates the status of the vehicle whether it is safe or not, and sends it to the VTA.

### 3.2 Implementation of the BVATS

Here, the implementation of the BVATS is discussed in different steps. Four smart contracts are developed to perform different cases in the BVATS architecture such as the authentication of the owner using the UVK, the authorization of the owner to add the AD's information, selecting the current driver, and destruction of the smart contract for removing AD's data. All things are handled by smart contracts automatically after getting request from the vehicle's owner.

The smart contract has an important role in this proposed architecture, where it is worked in the backend of the blockchain. The owner just calls any services through the owner interface and the smart contract acts in the backend as

per the owner's request. As shown in Fig. 2, the owner should be authenticated at least one time by using the *Authenticate* function that is associated with SC 1. Otherwise, the owner cannot access *Add Driver*, *Remove Driver*, and *Set Current Driver* functions. The owner can access all these functions on the owner interface. The vehicle owner can permit an AD by appending his data through the *Add Driver* function that is associated with SC 2. The owner can call the *Set Current Driver* function that is associated with SC 4 to permit the AD as the current driver and then the AD can drive the vehicle. Then, the information will be synchronized automatically with the vehicle's storage device. After completion of a driving session, the vehicle owner can remove the AD's data by using the *Remove Driver* function, which is associated with the SC 3, and finally updates the vehicle status.

The implementation process is composed in several steps, which are described in the following.

**Owner Registration:** A vehicle owner can submit the required information to the VSA, when buying a vehicle. Therefore, the VSA will send the owner and vehicle information to the VCA. The VCA will verify the information with the owner through the interface to ensure that the given information is correct. Then, The VCA will provide a login id, password, and owner interface link to the owner. Finally, the owner will be registered successfully to the blockchain network. Therefore, the owner can join the blockchain network.

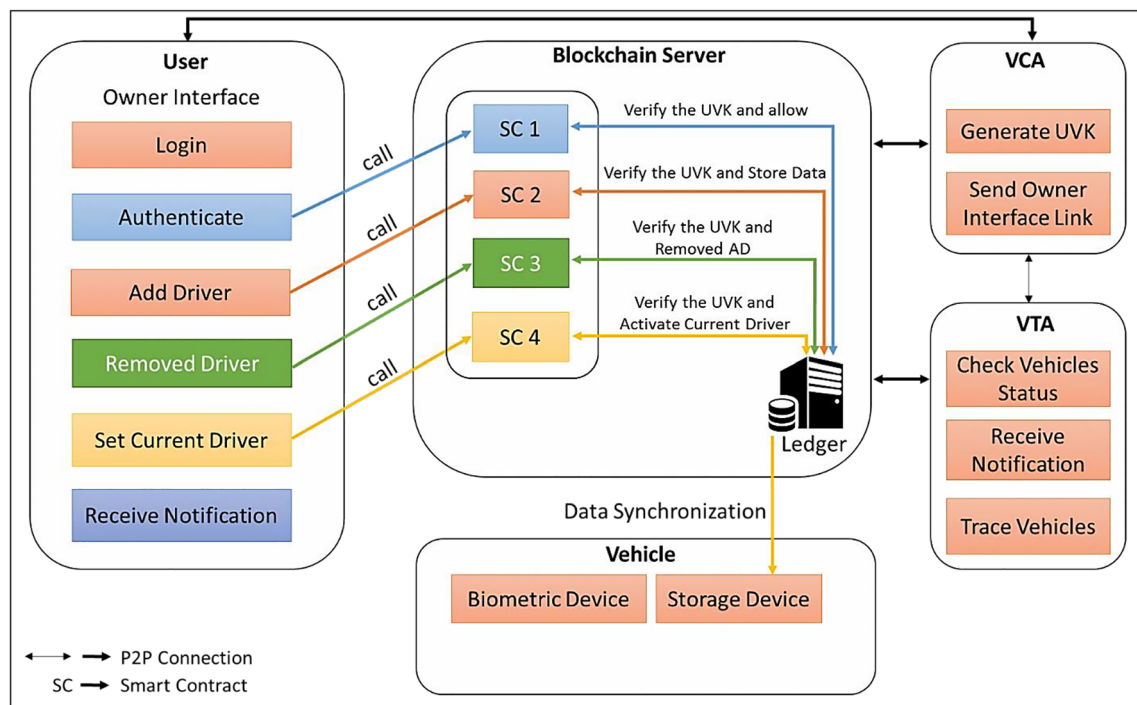


Fig. 2 Working procedure of smart contracts in the BVATS architecture

**Owner Authentication:** The UVK will be generated for the owner after verifying the owner's information. After completion of the registration, the owner requires to authenticate to the blockchain network. Thus, the owner needs to login to the interface and provides a user id and password to access the smart contract's facilities. Therefore, the smart contract will verify the UVK in the backend. If the UVK is associated with the owner's account address then it's correct. Otherwise, the authentication process will be discarded. Finally, the owner will be successfully authenticated to the network.

**Data Insertion:** The owner needs to add AD's information through a smart contract for driving the vehicle in every driving session. The UVK is must required to add AD's information. The smart contract only executes any event using the correct UVK. Therefore, the owner can receive a confirmation message of successful data submission. Then, AD will get permission for driving. After a driving session, the owner will destruct the smart contract requesting the self-destruct function.

**Driver Authorization and Unauthorized Access Detection:** Every driver has to authorize for driving in every driving session. Only the authorized person can drive an owner's vehicle. Before the drive, the owner has to set the current driver by calling the smart contract. The smart contract can be called from the owner interface. As per the owner's request, data can be synchronized with the vehicle device to verify the selected driver. Therefore, the selected driver can drive authorizing through the biometric device. An unauthorized access detection workflow diagram is shown in Fig. 3. The unauthorized access detection algorithm is also provided in the following:

#### Algorithm 2: Unauthorized Access Detection Algorithm

```

Input: Vehicle_Driver, Vehicle_speed, AD_biometric
1. Start
2. if(Vehicle_Driver == Owner)
   then
     a. then allow to drive;
   end
3. else if(Vehicle_Driver == AD && AD_biometric == true)
   then
     a. then Allow to drive;
   end
4. else
   a. If(Vehicle_speed > 0)
     then
       i. then send a notification to the owner and the VTP
       ii. And the VTP will be waiting for the vehicle status
     end
     b. else
       i. Don't need to notify;
     end
   end
5. End

```

**Notification Sending:** When an unauthorized person tries to access the vehicle, an waring alarm will be notified to the vehicle owner if the vehicle\_speed > 0. If the vehicle\_speed = 0, that means there is no risk with the vehicle. So, no notification will be sent.

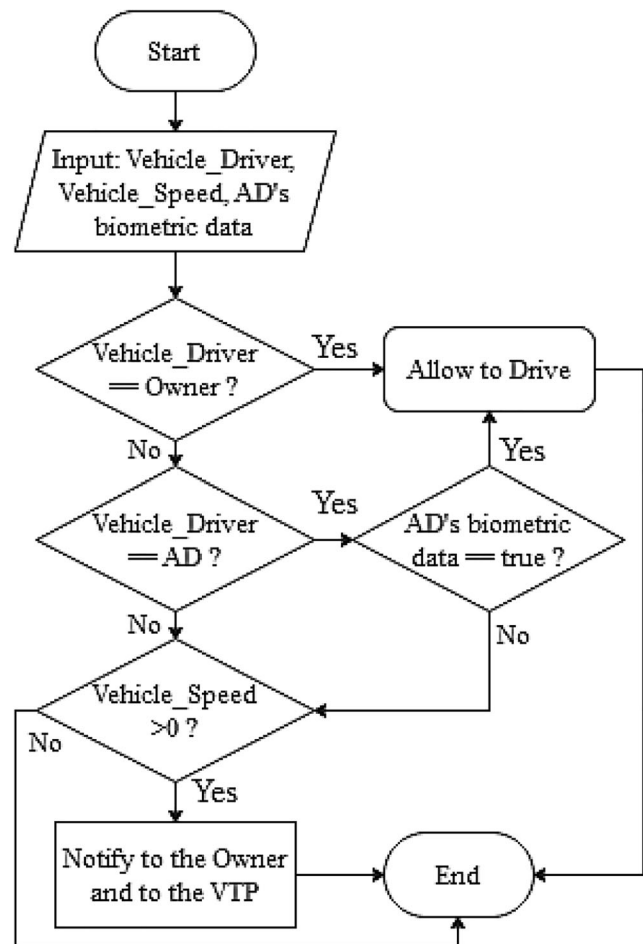


Fig. 3 Workflow diagram of the unauthorized access detection

## 4 Testing

This section describes the testing results of the implemented smart contract for the BVATS. We have experimented only with the proposed algorithm for the proposed framework. Smart contract code's have been written in solidity language using a web based platform named Remix IDE. The smart contract's code has been compiled and run successfully and, therefore, we got the results. Sometimes, we have needed to put input data such as UVK. for getting the desired output such as status of user authentication. Practically, smart contracts can get the input data automatically from the database.

**Environment Setup:** Experimental results are implemented by using the Remix-Ethereum IDE [45], which provides an interface for executing the smart contract. It is an open-source platform that can help us to write code of the smart contract in solidity language [46] in the browser. Solidity is a familiar language for writing a smart contract's code. It serves blockchain specific services with code, which runs in the EVM and supports testing, compiling, and

**Table 1** Functionality of used functions of the smart contract

Function Name	Used for
SetUVK	authenticating a vehicle owner
storedriverdata	adding an AD's data
CurrentDriver	setting the current driver
RemovesADs	removing the AD's data

deploying the smart contract. The preferred solidity compiler version is 0.5.17 and EVM version for the experiment is set to compiler default. The proposed algorithms are implemented in solidity with writing codes on the Ethereum platform. The selected environment to deploy and run the smart contract's code is Javascript VM.

In this work, the usage of the defined functions of the smart contract is shown in Table 1. Different scenarios of the experiment are given in the following:

**Smart Contract Deployment:** The smart contract has been deployed successfully from the account address `0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c` to the address `0x692a70d2e424a56d2c6c27aa97d1a86395877b3a`, which is shown in Fig. 4. In this experiment, all smart contracts code have been written within a single file (BVATS.sol) and compiled successfully.

**Owner Authentication by the UVK:** The owner needs to verify himself using the UVK when he needs to add data. So, the UVK is must require to verify the

owner. The UVK is set to *100* by calling the *setUVK* function. Here, it is randomly set to confirm the owner when AD's data will be added later. The *setUVK* function has only been taken for experiment purposes. In a practical scenario, this key will be available after the completion of the registration of the owner. Thus, this function is not required in a practical scenario.

As mentioned before that the UVK is required while the vehicle owner wants to add an AD's data to provide authorization for driving the vehicle. The *storedriverdata* function has been defined and used to add AD's data. The AD's data hasn't been added for providing a wrong UVK, which is shown in Fig. 5. No one can add data giving the wrong UVK. As it is known to the owner, he can only add the AD's data.

When the owner adds an AD's information, he has to provide the UVK at this time. Figure 6 is shown that the information has been added using the correct UVK. Here, the license id and name of the AD are taken *111* and *Sourav Banerjee*. Thus, the AD's data has been added successfully after verifying the correct UVK.

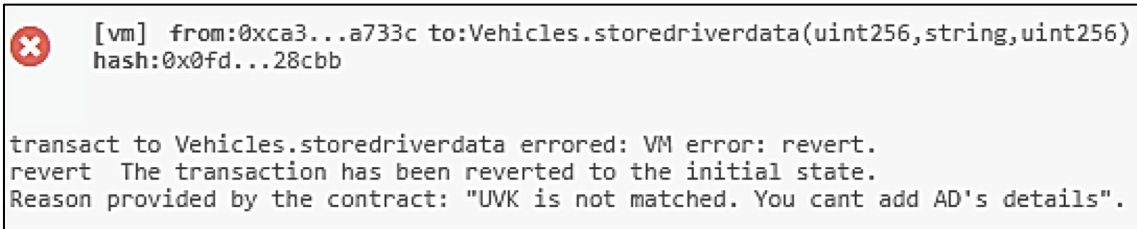
The data can't be added from another owner's account address. Even it's not possible to use the UVK *100* from another owner's account address. Figure 7 shows the transaction has been reverted as the UVK is trying to set UVK from another address `0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db`.

**Current Driver Selection** When the owner drives the vehicle, it's not required to call a smart contract to add an AD's data. As shown in Fig. 8, the current driver's name is randomly set to *Debashis Das* with his license Id *100,001* before adding the AD's information. Here, it is

status	0x1 Transaction mined and execution succeed
transaction hash	0xececb3a51541bedae8103fba6f77b9de99ed643d3cdceea1b61cf74b1dd0e307 
contract address	0x692a70d2e424a56d2c6c27aa97d1a86395877b3a 
from	0xca35b7d915458ef540ade6068dfe2f44e8fa733c 
to	Vehicles.(constructor) 
gas	3000000 gas 
transaction cost	773392 gas 
execution cost	550444 gas 
hash	0xececb3a51541bedae8103fba6f77b9de99ed643d3cdceea1b61cf74b1dd0e307 
input	0x608...10032 
decoded input	{ } 
decoded output	- 
logs	[ ]  
value	0 wei 

**Fig. 4** Smart contract has been deployed successfully





**Fig. 5** Data hasn't been added for giving the wrong UVK

assumed that the owner's *name* and *license Id* are already stored in the ledger. The *CurrentDriver* function has been used to set the current driver for a driving session. Only the owner can drive the vehicle if no AD's data has been added.

After adding an AD's data, the owner needs to set the AD as the current driver of the vehicle using the *CurrentDriver* function whenever it is needed. The current driver *Sourav Banerjee* is got permission for driving the vehicle in a driving session. Transaction for the current driver setting is shown in Fig. 9.

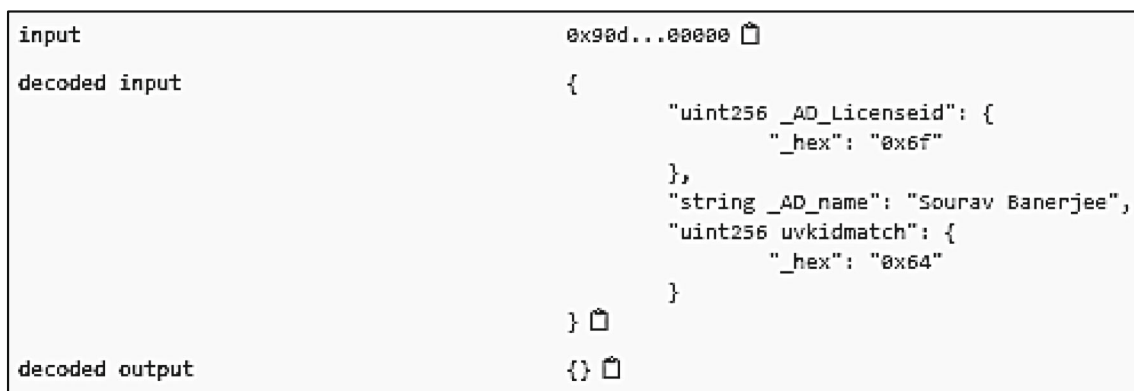
**Data Destruction** The smart contract can be destructed using the self-destruction function. Thus, the data cannot be accessed without calling it again. After calling the *RemovesADs* function, it's not possible to access an AD's data. Thus an unauthorized person cannot access the vehicle. Figure 10 shows the transaction after the destruction of the smart contract. Then, no information can be accessed through the smart contract. However, to access it again, the owner should call it and an automatic smart contract will be deployed, and then the owner can access it.

## 5 Performance analysis

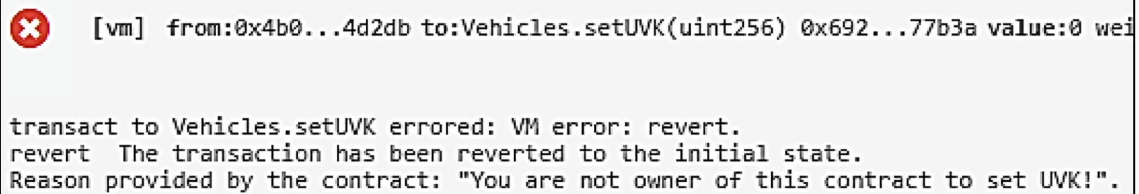
GSM based anti-theft system can be hacked by a hacker [21], IoT based anti-theft system has the possibility of cyber-attacks [22] and the possibility of attacks by a hacker can be presented in mobile communication based anti-theft system [20]. On the

other hand, blockchain's data almost cannot be altered by a hacker [48]. There are public and private keys to all blockchains. Since they are comprised of using cryptography, making the right combination of both keys, users can access the blockchain only. Thus, the proposed framework has a lot of advantages than other anti-theft systems. Each connected node in the blockchain network has the same copy of the information. However, data availability, data privacy, and data immutability are presented in the BVATS architecture [47].

Implementation of the BVATS and the smart contract's experimental result make ensure that unauthorized access can be detected by using the proposed vehicle antitheft system. So, it is clear to us unauthorized access wouldn't be happened, because authorization is required for driving a vehicle in every driving session. After a driving session, the AD's authorization information will be removed by the vehicle owner to restrict unnecessary access again. For the next driving session, the owner has to call the smart contract and add the AD's information. In existing systems, only one person's verification data can be stored in the vehicle. But, in this proposed system, multiple people can drive the vehicle authorized by the vehicle owner only. A biometric device is a secure identification and access control tool. This tool employs automatic methods to validate or recognize a living person's identity depending on physiological or cognitive properties. These include thumbprints, facial representations, retina, and appreciation of the voice [49]. Thus, the safety of the vehicle can be achieved in this proposed framework.



**Fig. 6** Data has been added successfully using the correct UVK



```
[vm] from:0x4b0...4d2db to:Vehicles.setUVK(uint256) 0x692...77b3a value:0 wei
transact to Vehicles.setUVK errored: VM error: revert.
revert The transaction has been reverted to the initial state.
Reason provided by the contract: "You are not owner of this contract to set UVK!"
```

**Fig. 7** The UVK cannot be added from another account address

As an owner is a blockchain node, he has a blockchain account. The UVK is associated with the owner blockchain account. Whenever he wants to request for any purpose, then at first, he has to authenticate as an authorized owner of this account. Therefore, he can able to request from his account. The UVK will be worked only for his account. The UVK cannot be checked from another owner's account as it isn't associated with another account. Thus, the UVK is truly confidential as a unique identification of the vehicle as well as the vehicle owner. In this research work, a UVK is used internally as a unique identification to verify the authentication of an owner, who wants to add some information to the blockchain ledger. The UVK key is completely inaccessible by any external entities. It will only be accessed by the smart contract, which will authenticate a genuine owner using the Blockchain's intrinsic security measures. Thus, the proposed system is a secured for vehicle theft detection scheme.

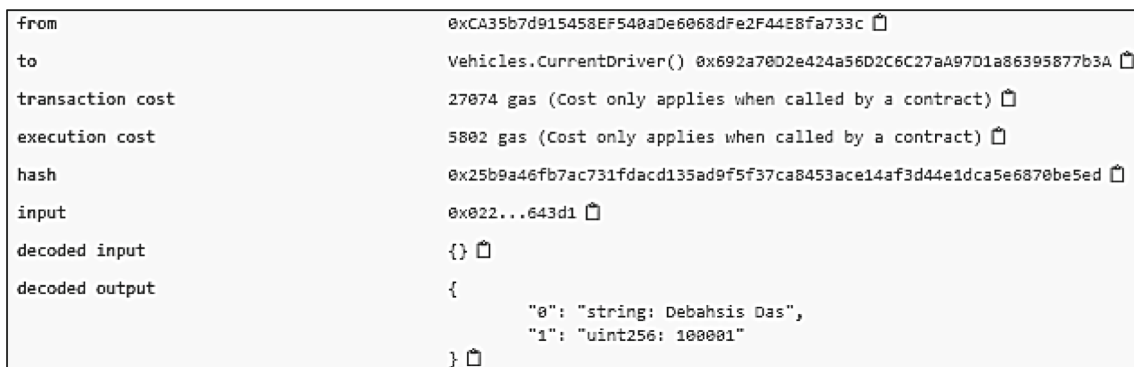
Blockchain's beneficial features can be achieved by using this framework as it is designed by utilizing blockchain technology. The BVATS provides an adequate resolution to avoid existing systems' issues. The proposed architecture provides a secure key management mechanism that is not available in existing systems [50]. The BVATS also delivers the difficulty from the stealing of personal information as it is acquired using the blockchain technology that can enable security of vehicles and privacy of the owner's information in the proposed architecture [23]. The smart contract is also secured in the blockchain ledger from unauthorized modification. Since, the smart contract can be used to access the stored data and to store processing data, thus it is stored on the blockchain ledger, which is a secure platform. No one can modify the smart

contract's code, as the blockchain can provide an immutable ledger. Thus, the smart contract is secured in this proposed BVATS architecture. So, the data stored in the blockchain ledger is nearly impossible to mangle [23]. And, the security of the framework can be enhanced the vehicle safety.

The performance of BVATS architecture is better than the existing methods concerning different characteristics are shown in Table 2. We have prepared the table after comparing it with the mentioned existing methods [17–19] based on the applied methodology/technology of these methods. We found out limitations of the existing methods, analyzed existing issues, and provided key features and benefits of the proposed BVATS method over the existing vehicle anti-theft methods.






**Malware Attack** Traditionally, centralized servers are attacked by attackers. They mainly used malware for stealing personal or business information. Attackers used phishing, spam, and driver by download methods to spread malware for targeting computers. Domain Name System (DNS) has been a targeted area of many Denial-of-Service (DoS) attacks. Attackers can send false DNS messages, as DNS reply messages are not authenticated. Thus, a signature-based approach is necessary for data transmission, as it is impossible to mangle or catch up with the signature pattern by a malware pattern. A signature-based approach is available in the blockchain that is decentralized and designed using cryptography. Thus, malware attacks cannot happen in BVATS while other systems suffer.

**Leakage of Personal Information** Existing systems [17–19] can lead to leakage of personal information, as these have



```
from 0xCA35b7d915458EF540aDe6068dFe2F44E8fa733c
to Vehicles.CurrentDriver() 0x692a70D2e424a56D2C6C27aA97D1a86395877b3A
transaction cost 27074 gas (Cost only applies when called by a contract)
execution cost 5802 gas (Cost only applies when called by a contract)
hash 0x25b9a46fb7ac731fdacd135ad9f5f37ca8453ace14af3d44e1dca5e6870be5ed
input 0x022...643d1
decoded input {}
decoded output {
  "0": "string: Debahsis Das",
  "1": "uint256: 100001"
}
```

**Fig. 8** Current driver setting before adding the AD's data

input	0x022...643d1 
decoded input	{ } 
decoded output	{ "0": "string: Sourav Banerjee", "1": "uint256: 111" } 
logs	[ ]  

**Fig. 9** Current driver setting after adding an AD's data

been used smartphone applications to access the proposed services. Our proposed system has the encryption method to access smartphone applications. Thus, personal information cannot be accessed by attackers using malware.

**Data Security** Users never know about the manipulation or deletion of stored data in existing systems [17–19]. But, our proposed system has data security as it is designed using blockchain. Blockchain data is secure due to having a cryptography feature. Thus, the proposed system's data is secure.

**Adequate Key Management** Key is an important feature to identify an entity. Existing systems [17–19] have login credentials like user id and password instead of having a proper key feature. Our proposed system has an adequate key feature such as UVK. The UVK can identify the vehicle as well as the vehicle's owner.




**Transparency** Existing systems [17–19] haven't a transparent database due to having a centralized database. But, a decentralized database has a transparent view of stored data. Thus, the proposed system has higher transparency of data.

**Immutability** Existing systems [17–19] haven't data immutability features. Thus, data can be manipulated easily here. But, our proposed system has a data immutability feature, where no one can alter the stored data. If anyone make changes in blockchain data, then all nodes can view the changed information as each node has same copy of data and updated data automatically after a time period.

**Availability** Existing systems [17–19] can suffer from a single point of failure. If a centralized server crashes, then no one can assess data from it. But, our proposed system has a data availability feature. Here, if one node crashes then the other node has the backup, as each node connected in the blockchain network has the same copy of data.

## 6 Conclusion and future work

Blockchain technology is growing rapidly in various domains for its security and efficiency. It is a decentralized computation and information sharing platform that enables us to connect multiple authoritative domains where no one can trust each other to cooperate, collaborate, and coordinate with each other in an intelligent decision-making process. The protection of vehicles is essential by developing a secure and cost-effective system as the vehicle theft rate increased in many developed countries. While consistent security and privacy in a traditional information system can be impossible to obtain, blockchain may do so by the use of its intrinsic public key infrastructure to ensure security against fraudulent attempts to manipulate records. The blockchain is a solution to increase the security of the online application. In this paper, we have described and implemented a decentralized vehicle anti-theft system using blockchain and smart contracts. The implementation process, a set of experimental results, and comparison analysis have been done and conclude that the BVATS provides a transparent way to reduce the possibility of leakage of personal information and improves the vehicle's anti-theft

input	0x022...643d1 
decoded input	{ } 
decoded output	{ "0": "string: ", "1": "uint256: 0" } 

**Fig. 10** Data cannot be accessed after the destruction of the smart contract

**Table 2** Key features and benefits of BVATS over the existing vehicle anti-theft methods

Methods	Centralized			Decentralized
Characteristic	GSM and GPS with Fingerprint Verification [17]	based on the Internet of things[18]	with a smartphone application[19]	BVATS
Malware Attack	✓	✓	✓	✗
Leakage of Personal Information	✓	✓	✓	✗
Data Security	✗	✗	✗	✓
Adequate Key Management	✗	✗	✗	✓
Transparency	✗	✗	✗	✓
Immutability	✗	✗	✗	✓
Availability	✗	✗	✗	✓

✗ = Not Applicable, ✓ = Applicable

system's safety and data security with a proper key management mechanism. Every node participating in the proposed system's Blockchain network can perceive the vehicle activities. So there is no possibility of any fraudulent activities. Thus, Blockchain can provide a significant contribution to the vehicle anti-theft system.

In the future, it would be addressed how can be developed security issues in automated vehicles applying blockchain technology. The following could be amended with the BVATS by developing more features.

- 1) Vehicle ownership can be changed using the smart contract.
- 2) An efficient solution can be acquired to control the vehicle speed using the smart contract.
- 3) We want to design a vehicle-to-vehicle (V2V) communication system by extending the proposed system for avoiding road accidents using the blockchain, multimedia, and sensing device.

## References

1. Nagaraja BG, Rayappa R, Mahesh M, Patil CM, Manjunath TC (2009) Design & Development of a GSM Based Vehicle Theft Control System. 2009 International conference on advanced computer control. Singapore 148–152. <https://doi.org/10.1109/ICACC.2009.154>
2. Ernst R (2018) Automated Driving: Cyber-Phys Perspect Comput 51(9):76–79. <https://doi.org/10.1109/MC.2018.3620974>
3. Nawa K, Chandrasiri NP, Yanagihara T, Komori T, Oguchi K (2012) Cyber-physical system for vehicle application. 2012 IEEE international conference on cyber Technology in Automation, control, and intelligent systems (CYBER). Bangkok 135–138. <https://doi.org/10.1109/CYBER.2012.6392540>
4. Becsi T, Aradi S, Gaspar P (2015) Security issues and vulnerabilities in connected car systems. 2015 international conference on models, and Technologies for Intelligent Transportation Systems (MT-ITS). <https://doi.org/10.1109/MTITS.2015.7223297>
5. Kim J, Jang JJ, Jung IY (2016) Near Real-Time Tracking of IoT Device Owners, vol 1085–1088. 2016 IEEE international parallel and distributed processing symposium workshops (IPDPSW), Chicago. <https://doi.org/10.1109/IPDPSW.2016.218>
6. Jiang R, Zhu Y (2019) Wireless access in vehicular environment. In: Shen X, Lin X, Zhang K (eds) Encyclopaedia of wireless networks. Springer, Cham. [https://doi.org/10.1007/978-3-319-32903-1\\_309-1](https://doi.org/10.1007/978-3-319-32903-1_309-1)
7. Yu K, Tan L, Shang X, Huang J, Srivastava G, Chatterjee P (2021) Efficient and privacy-preserving medical research support platform against cOVID-19: A blockchain-based approach. In IEEE consumer electronics magazine 10(2):111–120. <https://doi.org/10.1109/MCE.2020.3035520>
8. Chang Y, Lakovou E, Shi W (2020) Blockchain in global supply chains and cross border trade: a critical synthesis of the state-of-the-art, challenges and opportunities. International Journal of Production Research 58(7):2082–2099. <https://doi.org/10.1080/00207543.2019.1651946>
9. Singh PK, Singh R, Nandi SK, Nandi S (2019) Managing smart home appliances with proof of authority and blockchain. In: Lüke KH, Eichler G, Erfurth C, Fahrnerberger G (eds) Innovations for community services. I4CS 2019. Communications in computer and information science 1041:221–232. Springer, Cham. [https://doi.org/10.1007/978-3-030-22482-0\\_16](https://doi.org/10.1007/978-3-030-22482-0_16)
10. Zhu Y, Qin Y, Zhou Z, Song X, Liu G, Chu WC (2018) Digital Asset Management with Distributed Permission over Blockchain and Attribute-Based Access Control. 2018 IEEE international conference on services computing (SCC), San Francisco, pp 193–200. <https://doi.org/10.1109/SCC.2018.00032>
11. Zhu X, Wang D (2019) Research on Blockchain application for E-commerce. Finance Energy IOP Conf Ser: Earth Environ Sci 252: 042126. <https://doi.org/10.1088/1755-1315/252/4/042126>
12. Pawar MR, Rizvi I (2018) IoT based embedded system for vehicle security and driver surveillance. 2018 Second international conference on inventive communication and computational technologies (ICICCT). Coimbatore 466–470. <https://doi.org/10.1109/ICICCT.2018.8472984>
13. GGuo Z, Shen Y, Bashir AK, Imran M, Kumar N, Zhang D, Yu K (2020) Robust spammer detection using collaborative neural network in internet of thing applications. in IEEE Internet of Things Journal 2327–4662. <https://doi.org/10.1109/JIOT.2020.3003802>
14. Dashora C, Sudhagar PE, Marietta J (2019) IoT based framework for the detection of vehicle accident. Cluster Comput 23:1235–1250. <https://doi.org/10.1007/s10586-019-02989-z>
15. Yang F, Wang S, Li J, Liu Z, Sun Q (2014) An overview of the internet of vehicles. China Commun 11(10):1–15. <https://doi.org/10.1109/CC.2014.6969789>
16. Maglaras LA, Al-Bayatti AH, He Y, Wagner I, Janicke H (2016) Social internet of vehicles for smart cities. J Sens Actuator Netw 5(1):3. <https://doi.org/10.3390/jsan5010003>
17. Dey M, Arif MA, Mahmud MA (2017) Anti-theft protection of vehicle by GSM & GPS with fingerprint verification. 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE). Cox's Bazar 916–920. <https://doi.org/10.1109/ECACE.2017.7913034>
18. Liu Z, Zhang A, Li S (2013) Vehicle anti-theft tracking system based on Internet of things. Proceedings of 2013 IEEE International Conference on Vehicular Electronics and Safety.



- Dongguan, China 48–52. <https://doi.org/10.1109/ICVES.2013.6619601>
19. Shruthi K, Ramaprasad P, Ray R, Naik MA, Pansari S (2015) Design of an anti-theft vehicle tracking system with a smartphone application. 2015 International conference on information processing (ICIP). Pune 755–760. <https://doi.org/10.1109/INFOP.2015.7489483>
20. Yu K, Lin L, Alazab M, Tan, Gu B (2020) Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system. In IEEE transactions on intelligent transportation systems 1–11. <https://doi.org/10.1109/TITS.2020.3042504>
21. Gendrullis T, Novotný M, Rupp A (2008) A real-world attack breaking A5/1 within hours. Cryptographic hardware and embedded systems-CHES 2008. CHES 2008. Lect Notes Comput Sci 5154:266–282. [https://doi.org/10.1007/978-3-540-85053-3\\_17](https://doi.org/10.1007/978-3-540-85053-3_17)
22. Ko E, Kim T, Kim H (2017) Management platform of threats information in IoT environment. J Ambient Intell Humaniz Comput 9(4):1167–1176. <https://doi.org/10.1007/s12652-017-0581-6>
23. Alotaibi B (2019) Utilizing Blockchain to overcome cyber security concerns in the internet of things: a review. IEEE Sensors J 19(23): 10953–10971. <https://doi.org/10.1109/jsen.2019.2935035>
24. Ramaguru R, Sindhu M, Sethumadhavan M (2019) Blockchain for the internet of vehicles. Advances Comput Data Sci 1045:412–423. [https://doi.org/10.1007/978-981-13-9939-8\\_37](https://doi.org/10.1007/978-981-13-9939-8_37)
25. Aswathy SV, Lakshmy KV (2019) BVD - a Blockchain-based vehicle database system. Secur Comput Commun 969:220–230. [https://doi.org/10.1007/978-981-13-5826-5\\_16](https://doi.org/10.1007/978-981-13-5826-5_16)
26. Yi H (2019) Securing e-voting based on blockchain in P2P network. J Wireless Com Network 2019:137. <https://doi.org/10.1186/s13638-019-1473-6>
27. Das D, Banerjee S, Biswas U (2020) A secure vehicle theft detection framework using Blockchain and smart contract. Peer-to-Peer Netw Appl. <https://doi.org/10.1007/s12083-020-01022-0>
28. Banerjee S, Das D, Biswas M, Biswas U (2020) Study and survey on blockchain privacy and security issues. In Williams, I. (Ed.). Cross-industry use of Blockchain Technology and Opportunities for the Future 80–102. IGI Global. <https://doi.org/10.4018/978-1-7998-3632-2.ch005>
29. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang F (2019) Blockchain-enabled smart contracts: architecture. Appl Future Trends IEEE Trans Syst Man Cybernet: Syst 49(11):2266–2277. <https://doi.org/10.1109/TSMC.2019.2895123>
30. Yu S, Yang S, Li Y, Geng J (2018) Distributed energy transaction mechanism design based on smart contract. 2018 China international conference on electricity distribution (CICED). Tianjin. 2790–2793. <https://doi.org/10.1109/CICED.2018.8592130>
31. Felker D (2018) SELF DESTRUCTING SMART CONTRACTS IN ETHEREUM. <https://articlescarterio/blockchain/self-destructing-smart-contracts-in-ethereum/> Accessed 06 Dec 2019
32. Hyperledger (2019) Smart contracts and Chaincode. <https://hyperledger-fabric.readthedocs.io/en/release-1.4/smartcontract/smartcontract.html> Accessed 06 Dec 2019
33. Solidity (2017) Introduction to smart contracts. <https://solidity.readthedocs.io/en/v0.4.21/introduction-to-smart-contracts.html> Accessed 07 Dec 2019
34. Academy B (2020) What are smart contracts. <https://www.binancevision/blockchain/what-are-smart-contracts> Accessed 04 Jan 2020
35. Yuan R, Xia Y, Chen H et al (2018) ShadowEth: private smart contract on public Blockchain. J Comput Sci Technol 33:542–556. <https://doi.org/10.1007/s11390-018-1839-y>
36. Margo R (2020) Why use smart contracts to build Blockchain applications <https://dzone.com/articles/why-use-smart-contracts-to-build-blockchain-applic> Accessed 07 Jan 2020
37. Ethereum website (2019) Learn about Ethereum. <https://ethereum.org/learn/#ethereum-basics> Accessed 15 Dec 2019
38. Hlebiv O (2018) Ethereum smart-contract storage <https://applicature.com/blog/blockchain-technology/ethereum-smart-contract-storage> Accessed 30 March 2020
39. Solomon MG (2020) Ethereum smart contracts: tips for handling data in solidity <https://www.dummies.com/personal-finance/ethereum-smart-contracts-tips-for-handling-data-in-solidity/> accessed 30 march 2020
40. Wei J, Chiu C, Huang F, Zhang J, Cai C (2019) A cost-effective decentralized vehicle remote positioning and tracking system using BeiDou navigation satellite system and Mobile network. J Wireless Com Network 2019:112. <https://doi.org/10.1186/s13638-019-1436-y>
41. Mbarek B, Jabeur N, Pitner T, Yasar AUH (2019) MBS: multilevel Blockchain system for IoT. Pers Ubiquit Comput:1–8. <https://doi.org/10.1007/s00779-019-01339-5>
42. Qian M, Gao H, Liu W (2018) Android based vehicle anti-theft alarm and tracking system in hand-held communication terminal. 2018 IEEE international conference on consumer electronics-Taiwan (ICCE-TW), Taichung 1–2. <https://doi.org/10.1109/ICCE-China.2018.8448426>
43. Liu B, Liu N, Chen G, Dai X, Liu M (2018) A low-cost vehicle anti-theft system using obsolete smartphone. Mob Inf Syst 2018:16–16. <https://doi.org/10.1155/2018/6569826>
44. Liu Z, He G (2005) Research on vehicle anti-theft and alarm system using facing recognition international conference on neural networks and brain. Beijing. 925–929. doi: <https://doi.org/10.1109/ICNNB.2005.1614771>
45. Remix, Ethereum-IDE (2019) Welcome to Remix documentation. <http://remixethereum.org/> Accessed 05 Dec 2019
46. Solidity v0.6.0 (2019) Solidity. <https://solidity.readthedocs.io/en/v0.6.0/> Accessed 05 Dec 2019
47. Paik HY, Xu X, Bandara HMND, Lee SU, Lo SK (2019) Analysis of data Management in Blockchain-Based Systems: from architecture to governance. IEEE Access 7:186091–186107. <https://doi.org/10.1109/access.2019.2961404>
48. Zheng B, Zhu L, Shen M et al (2018) Scalable and privacy-preserving data sharing based on Blockchain. J Comput Sci Technol 33:557–567. <https://doi.org/10.1007/s11390-018-1840-5>
49. Kiruthiga N, Latha L, Thangasamy S (2015) Real Time Biometrics Based Vehicle Security System with GPS and GSM Technology. Procedia Comput Sci 47:471–479. <https://doi.org/10.1016/j.procs.2015.03.231>
50. Jing Q, Vasilakos AV, Wan J, Lu J, Qiu D (2014) Security of the internet of things: perspectives and challenges. Wirel Netw 20(8): 2481–2501. <https://doi.org/10.1007/s11276-014-0761-7>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Debashis Das** (Dept. of Comp. Sc. and Engg, University of Kalyani, India, debashis2124@gmail.com) is currently working as a university research scholar at Kalyani University. He has completed the Master of Technology in Computer Science and Engineering in 2018 from Kalyani Government Engineering College. He has completed a Bachelor of Technology in Computer Science and Engineering in 2015 from the

Government College of Engineering and Leather Technology. His research interests are including Cloud Computing, Internet of Things (IoT), Distributed Computing, and Blockchain Technology. Now, He is working on Blockchain technology.



**Uttam Ghosh** (Vanderbilt University, Nashville, USA, uttam.ghosh@vanderbilt.edu) received his Ph.D. and M.S. in Electronics and Electrical Communication Engineering from Indian Institute of Technology, Kharagpur. He completed B.Tech in Information Technology from West Bengal University of Technology. He is working as an Assistant Professor of Practice in EECS, Vanderbilt University, Nashville, Tennessee. His research interests

include Cyber Physical System Applications and Security, Software Defined Networking (SDN), Distributed & Mobile Computing, Wireless Sensor & Ad Hoc Networks, Internet of Things (IoT) and Content Centric Networking (ICN), Smart Grid, Cloud Computing, Wireless Networks, Protocol Stacks for Wireless/Wired Networks, Indoor Positioning System



**Sourav Banerjee** (Dept. of Comp. Sc. and Engg, Kalyani Govt. Engg. College, India, mr.sourav.banerjee@ieee.org) holds a PhD degree from the University of Kalyani in 2017. He is currently an Assistant Professor at Department of Computer Science and Engineering of Kalyani Government Engineering College at Kalyani, West Bengal, India. He has authored numerous reputed journal articles, book chapters and International confer-

ences. His research interests include Big Data, Cloud Computing, Cloud Robotics, Distributed Computing and Mobile Communications, IIoT, Blockchain. He is a member of IEEE, ACM, IAE and MIR Labs as well. He is a SIG member of MIR Lab, USA. He is an Editorial board member of Wireless Communication Technology. He is a lead guest editor of Complex and Intelligent Systems Journal (Springer, SCI, IF: 3.79) for the Special Issue on "Advancement and Trends in Green Cloud Computing, Blockchain and IoT for Modern Applications and Systems". A number of worldwide research scholars are attached with him.



**Utpal Biswas** (Dept. of Comp. Sc. and Engg, University of Kalyani, India, utpal0172@gmail.com) received his B.E, M.E and PhD degrees in Computer Science and Engineering from Jadavpur University, India in 1993, 2001 and 2008 respectively. He served as a faculty member in NIT, Durgapur, India in the department of Computer Science and Engineering from 1994 to 2001. Currently, he is working as a professor in the department of

Computer Science and Engineering, University of Kalyani, West Bengal, India. He has over 130 research articles in different journals, book chapters and conferences. His research interests include optical communication, ad-hoc and mobile communication, semantic web services, E-governance, Cloud Computing etc.



**Ali Kashif Bashir** (Department of Computing and Mathematics, E-154, John Dolton, Chester Street, M15 6H, Manchester Metropolitan University, Manchester, United Kingdom, a.bashir@mmu.ac.uk) is a Distinguished Speaker of ACM. His past assignments include an Associate Professor of information and communication technologies with the Faculty of Science and Technology, University of the Faroe Islands, Denmark; Osaka University, Japan; the Nara National College of Technology, Japan; the National