

Ushering in a IoV Ecosystem for Blockchain-based Identity and Trust Supervision System

Rachna Sharma

Student

Department of CSE

Motilal Nehru National Institute of Technology

Corresponding Author's E-mail: rachna.sharma@gmail.com

Abstract

Over the last decade, there has been tremendous growth in both the Internet of Things (IoT) and the Internet of Vehicles (IoV) from both academics and industry. Autonomous vehicles (AVs) are made up of a multitude of sensors, IoT devices, control units, gateways, and other components. As a result, the software of the various sensors and IoT devices must be updated to the most recent version by the Software Vendors. The purpose of this article is to offer a blockchain-based identity and trust management architecture for the IoV ecosystem, with the goal of providing secure software updates. The proposed framework is divided into two processes: identification and registration of ecosystem entities, and entity authentication, which is based on established W3C standards for defining verifiable credential presentation, verification, and revocation. In a case study, we demonstrate the viability of our strategy.

Keywords: *Internet of Things, Communication channel, W3C standards, Blockchain, Internet of Vehicles*

INTRODUCTION

The IoV ecosystem enables automobiles to establish a communication channel via which they can exchange data. This information interchange offers a number of advantages, including improved road

safety and traffic control. However, there are various vulnerabilities that can be exploited in IoV ecosystems, which might have negative consequences such as causing accidents. As a result, the validity of the data transmitted in the established

communication channel must be ensured. To that end, we provide our first contribution, an Identity and Trust Management Framework that allows stakeholders in the IoV ecosystem to be identified and authenticated while also securing their communication. The framework uses Blockchain technology to save the Identifiers of the stakeholders in a decentralised manner, removing the single point of failure while boosting availability and eliminating the need for a trusted third party, centralised Certificate Authority (CA) to manage the identifiers. The centralised system is more vulnerable to a single point of failure, and it makes data breaches easier in the event that the server containing the certificates is hacked. It's worth noting that the suggested framework adheres to the W3C's proposed decentralised identifiers standards. Furthermore, it employs Verifiable Credentials to foster trust among the IoV ecosystem's stakeholders. Credentials are a series of claims (or statements) made by one entity, the Issuer, for another entity, the Holder, in most cases (or Prover). Verifiable Credentials encase the above-mentioned assertions in a set of metadata and digital proofs (the most common sort of digital evidence is a digital signature) that cryptographically establish who is the Credentials' Issuer. Sensors, control units,

and IoT devices are installed in vehicles, and these devices must be maintained by authorised services via a secure communication channel. To that aim, we present our second contribution, which is the application of the aforementioned Identity and Trust Management Framework to an important use case in the IoT area, namely, the safe and private software updating of IoT devices installed in vehicles.

The rest of the paper is organised as follows. The background on blockchain technology and similar efforts is presented in Section II. In Section III, we outline I the blockchain platform that will be employed after examining its suitability for our IoV ecosystem requirements and comparing it to other potential blockchain platforms, (ii) its stakeholders and their roles, and (iii) its processes. Finally, in Section IV, we wrap up the study and outline the work's next steps.

BACKGROUND & RELATED WORK

A. Blockchain Technology

Cryptography and, more specifically, hash functions tie data blocks together in blockchain technology and enable data integrity checks. To achieve the appealing guarantees, consensus protocols, timely consecutive cryptography, and a

decentralised network design are used to enhance the data linking notion. The assurances that enable secure and reliable data bookkeeping include ledger data immutability (tamper resistance), integrity, traceability, transparency, and availability.

The consortium-based Distributed Ledger Technology (DLT) is one of the branches that emerged from blockchain research. The biggest distinguishing feature of consortium-based DLT, in comparison to public blockchains, is that only approved nodes participate in the network. This constraint allows the network to use voting-based consensus protocols like Practical Byzantine Fault Tolerance (PBFT), which improves the network's scalability. As a result, consortium blockchains obtain faster transaction finality and shorter transaction verification times.

In the IoV, Identity Management (IdM) is one of the options that need the benefits of DLT. Furthermore, a consortium-based DLT approach is required for the IoV with numerous stakeholders as infrastructure participants. Vehicle insurance and the legal requirement of technical inspection of automobiles from trusted associations in order to get, for example, a car licence and number plate are the reasons for this.

Multiple recognised authorities contribute to the admittance of a vehicle in such processes. As a result, the ledger between the authorities is reliant on data from numerous approved sources. To minimise trust barriers between consortium partners, external input in the form of transactions from authorised participants should pass a fair consortium decision. Following that, DLT's tamper-proof bookkeeping property eliminates the need for a third party to manage bookkeeping amongst stakeholders.

External input verification in closed environment DLT frameworks is cutting-edge research that necessitates careful consideration during application development. Furthermore, the emphasis on privacy-preserving strategies, scalability solutions, and blockchain kinds (public, private, and consortium blockchain networks) drives distributed ledger framework development to customised design choices. Different DLT designs take distinct approaches to the aforementioned issues and provide different benefits and drawbacks. As a result, when it comes to designing IdM, choosing the correct DLT framework is critical. In our situation, IoV-specific criteria as well as the choice of independent, federated, or self-sovereign

identity were important considerations (SSI). In section III-A, IdM techniques dominate our framework selection.

B. Related Work

In this section, we discuss existing research on I addressing security in Internet of Things (IoT) software/firmware upgrades and (ii) IdM leveraging Blockchain technology.

The authors proposed a blockchain-based approach among AV makers that uses smart contracts to assure the validity and integrity of AV firmware updates. In comparison to centralised third-party services, AVs participate in the firmware update distribution process for high availability and fast delivery.

Instead of relying on a centralised vendor network, Lee & Lee et al. developed a blockchain-based method to secure the integrity of firmware upgrades for embedded IoT devices. To secure the integrity and versionability of updates, a peer-to-peer file sharing network such as BitTorrent is proposed for dissemination.

Yohan and Lo et al. developed a blockchain-based system in that intends to allow safe verification of device manufacturer firmware. The suggested

system is based on the Ethereum architecture and leverages smart contracts.

A decentralised blockchain IdM platform called Ockam network is being created to provide interoperability, security, privacy, trust, and reliability among IoT devices. It uses open standards for decentralised identifiers (DIDs) and verified credentials defined by the World Wide Web Consortium (W3C) and the Decentralized Identity Foundation (DIF) to assure identity compatibility. The platform is made up of operations including registration, authentication, and permission between IoT ecosystem actors.

In comparison to the previous works, we differ in that I we focus on the IoV ecosystem, (ii) we require/combine identity management as a requirement for our secure software update use case, (iii) we also address the issue of credential revocation, which is critical in advanced identity management solutions, and (iv) we map the stakeholders of our framework to the European Union's standardisation registration paradigms (EU).

PROPOSED IDENTITY & TRUST MANAGEMENT FRAMEWORK

A comparison of DLT frameworks and an identification technique of IoV stake-

holders are used to develop the suggested identity and trust management framework for secure communication. The parts that follow describe this procedure by outlining the criteria for selecting a DLT framework in the first place. Following is a comparison of DLT frameworks that identifies Indy as the best fit for the IoV use case. The Infrastructure Stakeholders section defines and maps stakeholders to Indy's identity roles. This method is required for determining the logical process of the IdM use case for secure communication.

A. Blockchain Platforms Evaluation

The evaluation criteria for selecting a DLT framework for IdM are summarised in the bullet points below.

- Existing identity management ideas
- Needed identity management concepts
- Existence of privacy-preserving strategies
- Extensibility of the framework architecture

The IoV situation necessitates a consortium-based DLT system with scalability and real-time capability to handle the position and velocity dynamics that occur in the IoV, as indicated in section II-A.

Table I DLT Framework Comparison in Respect of Identity Management in the IOV

DLT Framework	IdM Support	Privacy	Expandability
<i>Sawtooth Fabric</i>	Medium Medium (Structure of	Low Medium (Channels)	High High
<i>Indy</i>	High	High	High (Plugins)

1) Framework Design: The transaction client and processor are the most important parts of the Hyperledger Sawtooth framework. They form a transaction family that allows you to change the ledger state. Transaction clients create and sign transactions before sending them to the transaction processor for approval. Validator nodes are responsible for transaction processor assignments and submitting properly verified transactions to the sawtooth journal component, which is in charge of appending transaction blocks to the ledger.

Clients register at Fabric authorities in the Hyperledger Fabric system, which allows them to find Fabric peers. Clients then launch signed transaction proposals, gather peer signatures, inspect responses, and eventually broadcast transactions to an ordering service. The ordering service checks signatures, orders transactions, produces blocks, and distributes blocks to

ledger peers who append them to the ledger. Fabric is a scalable idea since it allows additional Fabric networks to be connected.

When compared to Sawtooth, the Hyperledger Indy architecture uses a similar trans-action concept. The transaction client is the Indy client System Development Kit (SDK), which submits preconfigured transaction requests. All critical ledger interactions regarding Decentralized Identifier (DID) maintenance are covered by indy ledger transactions. The Indy transaction repertoire includes DID generation and querying, key rotation, credential schema construction, credential definition creation, and so on. Additionally, the Indy SDK uses wallets to allow local credential management. Wallets can be used to store credentials, cryptographic key pairs, and to cache Decentralized Identifiers (DIDs).

All frameworks use scalable voting-based consensus algorithms like PBFT to achieve high transaction throughput.

2) Identity Management Suitability: A identification transaction family is provided by the Sawtooth framework. Client roles and permissions (policies) are implemented by the identity transaction

family, which allows authorisation. Additionally, cross-transaction family namespace constraints can be be up. Maintenance and revision design are two capabilities that IdM lacks. The system designer is in charge of devising privacy-enhancing methods for verifying identities.

Clients can use fabric channels to access specific chain information. As a result, information separation/access is possible, enhancing data privacy capabilities. Fabric also helps clients to protect their data from being exposed on the global blockchain storage. Clients can instead use their own private storage. Fabric allows clients to be authorised at the basic authorities that grant access to Fabric peers.

Indy's verifiable credential capabilities allow many private users to be linked to public credential schemata and authorities, resulting in cryptographically sealed trust relationships. With this, trustworthy identity management procedures like registration, authentication, authorization, and revocation can be implemented amongst clients who do not maintain their DIDs in a publicly accessible ledger. Furthermore, even though such flows are optional, Indy provides transaction flows that cover legal subtleties. Transaction Author Agreements (TAAs), for example,

attach explicit confirmation information to transactions. From a technical standpoint, such explicit agreements improve opportunities for adding private information to the immutable ledger. The TAA AML (Transaction Author Agreement Acceptance Mechanisms List) stores acceptance mechanisms in the Indy nodes' configuration ledger for this purpose. Indy plugins are similar to Sawtooth's transaction family concept in that they allow for the construction of customised transactions with appropriate validations. Plugins could be used to add missing IdM extensions. Designing dynamic validations that allow consortium members to actively interact during transactions, for example, could enable consortium-based transaction data verification and, as a result, consortium-only control. For the construction of advanced IdM solutions, Indy represents the platform with the most out-of-the-box functionality.

Table I summarises the comparison and illustrates Indy's suitability for advanced IdM, which is the evaluation's conclusion.

B. Identified Stakeholders and Processes of the Framework

The system's recognised stakeholders and their roles are presented in this section.

In order to identify the parties engaged in the registration of automobiles to the system, we used the paradigm offered by the European Union. In III-B, you'll find a list of the identified stakeholders. Table II highlights the parties engaged in the use case depicted in Figure 1 below. Although we used the European Union paradigm to identify the stakeholders of our system in a tangible and realistic way, we will only discuss the role of the stakeholders who engage in the Use Case shown in Figure 1 below in the rest of this Section.

Table II Framework Stakeholders

Stakeholder	Hyperledger Indy	Ecosystem Role
-	Trustee	Steward creator
-	Steward	Trust Anchors Cre-
Government	Trust Anchor	Issuer of Credential
Vehicle manufacturer	Trust Anchor	Issuer of Credential
Pre-market device soft-	Trust Anchor	Issuer of Credential
Pre-market device soft-	User	Prover/Verifier
Post-market device	Trust Anchor	Issuer of Credential

Post-market device	User	Prover/Verifier
Vehicle Owner or Transportation Authority (Vehicle)	Trust Anchor	Issuer of Credential Definition
Vehicle Registration	Trust Anchor	Issuer of Credential
Insurance Company	Trust Anchor	Issuer of Credential
VAT payee / tax office	Trust Anchor	Issuer of Credential
Roadworthiness	Trust Anchor	Issuer of Credential
Vehicle	User	Prover/Verifier

1) Role of the Trustee and the Steward:

Trustees and Stewards are special positions provided to users in Hyperledger Indy who play a vital role in establishing confidence in the network. However, in the paradigm that we follow, this is not the case. The system is bootstrapped by assigning/binding the roles of Trustees and Stewards to real users who can truly build trust in the system due to their inherent properties; the bootstrapping of the system by assigning/binding the roles of Trustees and Stewards to real users who can truly build trust in the system due to their inherent properties is a matter of future work. Trustees create users with the role of Stewards, and Stewards create users with the role of Trust Anchors 1; the latter have the ability to write transactions to the Ledger, for example, to publish to the Ledger Decentralised Identifiers and to create and publish to the Ledger Credential Schemas and Credential Definitions. Each Steward also serves as a Trust Anchor.

By assigning the Position of Trust Anchor to the Government, the Pre-market device software providers, and the Post-market device software vendors, we presume that a Trustee has created a user in the system with the role of Steward who is responsible for onboarding to the System. To accomplish so, the Steward must first establish secure communication with each of the aforementioned stakeholders. This entails the development of two distinct pairwise DIDs, one for the Steward and one for each Stakeholder. In addition to the DIDs, a verification key and a signing key are generated. These pairwise DIDs are private data held privately within software applications managed by the Steward and Stakeholders, referred to as "wallets."

A mutual authentication technique is used to establish secure communication, in which both parties must show to each other that they have the signature keys

matching to the pairwise DIDs. The messages transferred are encrypted with the verification keys of the two parties after the mutual authentication is completed successfully, ensuring the confidentiality of the exchanged messages. The Government then creates a DID and sends it to the Steward with its verification key; the Steward then publishes the DID and the verification key to the ledger and assigns the function of Trust Anchor to the Government. All data published to the Ledger is accessible to all network users with at least read access, and they are assumed to be trustworthy by all network users. Vendors of pre-market and post-market device software will follow the same method.

The process of I establishing a secure connection between a Trust Anchor and a new user, (ii) the Trust Anchor's subsequent write of the DID and verification key to the ledger, and (iii) the assignment of a role to the new User will be referred to as Registration of the user to the system in the rest of this paper. Additionally, the process of producing unique DIDs (together with the related verification and signing keys) is referred to as Identification 2. Table III below shows a list of the system's processes.

Table III Framework Processes

Process ID (PID)	Process Name
1	Identification and Registration
2	Authentication (VC Presentation, VC Verification, VC

2) Role of the Government:

A Steward (or Trust Anchor) with the job of Trust Anchor should first register the government in the system. Then it creates and publishes Credential Schemas to the Ledger that can be used as 'standards' for credential issuance by pre-market and post-market device software developers.

3) Role of Pre-market and Post-market device software vendor:

Pre-market and post-market device software vendors design and manufacture devices such as video cameras, lidar, radar, odometers, GPS, inertial measurement units (IMUs), and embedded electronics such as Electronic Control Units (ECUs) that are embedded to the vehicle before or after it is transferred to an Owner or Operator, respectively. They are both in charge of the software updates that are loaded on the devices on a regular basis.

Credential Definitions for Pre-market and Post-market device software vendor APIs

are issued by the Pre-market and Post-market device software vendors, respectively. These Credential Definitions are based on the Government's Credential Schemas, which were released to the Ledger.

4) The Device Software Vendor API's Pre-Market and Post-Market Roles:

Pre-market device software vendor APIs are Cloud-based APIs that comprise services and endpoints for delivering device software upgrades. The following are the phases in the process of the Vehicle authenticating the API:

1. The API connects to the Vehicle and creates a secure connection.
2. The Vehicle issues a Proof Request for Credentials pertaining to the types of software for which the API is authorised to offer updates.
3. The API pulls the relevant Credentials from its wallet, generates a Credential Presentation with a Proof that the Credentials were granted by the device software provider, and delivers the Presentation to the Vehicle.
4. The Vehicle checks the Proof provided in the Credential Presentation regarding the Credential Issuer, i.e. the Software Vendor, using the Ledger.

The procedure is depicted in Figure 1. The method described above is only half of mutual authentication; because mutual authentication is required, the Vehicle must additionally provide Credentials to the API proving that the device to be updated is one of the devices made by the Software Vendor doing the update. The software vendor is the issuer of these credentials. The API checks the Proof given in the Credentials about the Issuer of the Credentials, i.e. the Software Vendor, using the Ledger. Because the Software Vendor's DID and verification are published and resolvable from the Ledger via a Trusted Anchor, both entities trust it.

Following successful mutual authentication, the API shall return to the Vehicle requested Credentials including the digital fingerprint (e.g., a hash) of the Code matching to the most recent software update to be loaded on the device. The device software vendor will be the issuer of these credentials. By querying the Ledger, the Vehicle will validate the Credentials proof, and then examine the Integrity of the code that is delivered to it using a technique similar to that used to verify digital signatures.

If the software code Integrity check fails, it indicates that the code has been tampered

with or that the code does not belong to the Pre-market Software Vendor.

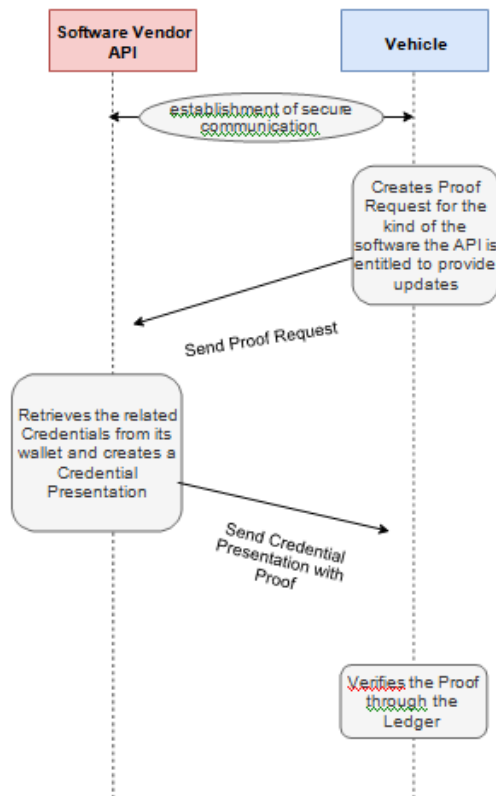


Figure 1: The Vehicle's Authentication of the Software Vendor API

The software update will not be installed by the vehicle. The API Credentials will be revoked when a secure link between the Vehicle and the Software Vendor is established. 3. The Post-market device software vendor API serves the same purpose as the Pre-market device software vendor API.

CONCLUSION AND FUTURE WORK

Because of the availability of sophisticated IdM principles of SSI, framework

expandability, and privacy-preserving approaches, we chose Hyperledger Indy as our DLT framework for the IdM in the IoV. It is feasible to design DID issuance and management across IoV infrastructure stakeholders using Indy's IdM capabilities.

The safe software update scenario between a car and a cloud api provides particular ecosystem responsibilities for stakeholder identification. Furthermore, automotive standardisation formalities introduce necessary authority for a valid IdM design. The incorporation of all identities into the DID credential architecture creates DLT-based credential links between identities that are connected to PKI keypairs. The safe exchange of software updates is enabled via mutual authentication of credentials and associated PKI keypairs.

REFERENCES

1. M. Baza, M. Nabil, N. Lasla, K. Fidan, M. Mahmoud, and M. Abdallah, "Blockchain-based firmware update scheme tailored for autonomous vehicles," 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakesh, Morocco, pp. 1–7, 2019.

2. S. Y. Lim, P. T. Fotsing, A. Almasri, O. Musa, M. L. M. Kiah, T. F. Ang, and R. Ismail, "Blockchain technology the identity management and authentication service disruptor: a survey," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 4-2, p. 1735, 2018.
3. B. Curran and J. Richards, "What are oracles," *Smart Contracts, Chainlink & "The Oracle Problem"*. Accessed, vol. 15, 2019.
4. K. Olson, M. Bowman, J. Mitchell, S. Amundson, D. Middleton, and C. Montgomery, "Sawtooth: An introduction," *The Linux Foundation*, Jan, 2018.
5. F. Masood and A. R. Faridi, "Distributed ledger technology for closed environment," in *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE, 2019, pp. 1151–1156.
6. A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 839–858.