Secure Distributed Network Model to Store Vehicle Transaction Records Through Blockchain Platform

Juan Carlos López-Pimentel
Universidad Panamericana
Facultad de Ingeniería
Zapopan, Jalisco, México
clopezp@up.edu.mx

Miguel Alcaraz Rivera
Universidad Panamericana
Facultad de Ingeniería
Zapopan, Jalisco, México
malcaraz@up.edu.mx

***Abstract*—Blockchain is an emergent technology applied initially to cryptocurrencies, but with a much further reach. We present a secure, transaction-based system that handles motorized vehicle provenance (including ownership) over a distributed model using smart contracts on a blockchain. The system model includes the specification of various security protocols, verified using AVISPA-SPAN tool and guaranteeing privacy, authenticity, and other security features. A client of the system can have different roles accordingly to the relationship it has with the vehicle, and it can get various services and request transactions accordingly. Our proposal might contribute to reduce fraud and expand the applications of Blockchain.**

***Index Terms*—Smart contract; Smart property; Blockchain and Cloud Computing.**

I. Introduction

Due the high value of automotive vehicles in general, the selling process is usually subject of constant scrutiny, but one that sometimes is as simple as the transference of possession and a physical signing of a notice of transfer. This process is an easy subject to fraud, causing harm to people who may expend a big fraction of their income in those transactions. We have detected some specific application problems when a combination of electronic and physical documents are introduced, as it adds a lack of clarity in the process of changing ownership.

A novel way to defend against this kind of frauds and secure the transactions is the addition of enforced smart property technologies. A smart contract is a computer program intended to digitally facilitate, verify, or enforce the negotiation and performance of a contract, using transactions. These transactions are traceable and irreversible.

In this paper, we present a secure distributed model solution based on smart properties technology to handle automotive related transactions, including vehicle ownership. The transactions can be processed, validated and updated through a secure and useful system with traceable and irreversible characteristics. Our model is presented as a set of security protocols, each of them shown in Alice and Bob notation and verified formally using AVISPA-SPAN tool.[1]

Our model includes different roles that a user can acquire accordingly to the relationship it has with the material vehicle and it can get services and request transactions from a Vehicle Blockchain Network (*VBN*) through a *Server wallet*.

The paper is organized as follows: Section II covers recent technologies that can handle system such as blockchain, smart contracts and smart properties. Then in Section III, we describe a case study emphasizing a problem in vehicle ownership transactions and we propose a model through a Blockchain network. Sections IV, V and VI explain in detail the model we are proposing. Finally, our conclusions are exposed.

II. Related Work

Blockchain emphasizes the following desirable features: Immutability, distributability and traceability, [1]. These features are reason enough by itself to choose a blockchain over other solutions and reason for which new blockchain-based solutions have arisen. The existence of smart contracts over a blockchain expand the use cases of this revolutionary technology above its original use in Bitcoin. As alternative to Bitcoin, other cryptocurrency systems are continuously in development, and in some cases, their network also have processing power, to be consumed as a service for the users, [2]. Although complete autonomy was not yet implementable due to the nonexistence of a sufficiently mature smart contract environment, [3], or in some cases, was impossible at all [4], [5]. Now, it seems to be different, among multiple platforms that allow using Blockchain with smart contracts to model and run full autonomous environments, like CoinDesk, [6] and Ethereum, [7], among others.

Blockchain provenance is not limited to information storage. Smart contracts inside the transactions allow for programmable interactive environments [8] where the blockchain will be enriched with valid data and binding transactions every time something interesting happens to the physical part of the object.

Some works are strongly related to ownership, e.g., in digital art [9], where the Bitcoin blockchain is used to identify and authenticate ownership of digital artwork property. Recently, [10], authors try to solve the problem of handling photograph ownership and authorization of fair use using smart contracts over the Ethereum blockchain. The patent [11], explains a more generalized solution to record ownership

[1]The SPAN specification can be downloaded via https://github.com/jclopezpimentel/BlockChainVehicles/tree/master/spanProof

978-0-7381-0495-9/20/$31.00 ©2020 IEEE
DOI 10.1109/Blockchain50366.2020.00052

over a blockchain; this solution involves some entities similar to our proposal, where *a digital wallet* access can be used to verify ownership information. Ownership solutions of Artificial Intelligences (AI) are also proposed, [12]. As in those cases described before, our work is focused on ownership, specifically of vehicles and tries to extend the application area of Blockchain, describing in detail a distributed network providing some specially useful security characteristics.

III. Outline

We expose a case study related to a private automotive transaction system, finding a flaw where malicious intent could be used to attack the system. Then, we outline a comprehensive proposal, including some stages in a blockchain technology platform.

A. Case Study

Although the following case is focused on Mexico, the fraud problem applies to various countries around the world, [13], [14]. Before 2014, the legal procedure steps of selling a used vehicle in Mexico are: 1) The buyer might check that no legal problems are involved with the vehicle; an online check can be used introducing the plate number; 2) The buyer makes the car payment to the seller. 3) The seller endorses his bill to the new owner and gives possession of the car to the buyer. 4) The change of ownership is notified to the city government. In some places a change of plates is necessary and a new set of plates is given to the new owner.

From 2012, Mexico introduces electronic signatures,[2] so from 2014, paper bills stopped being a legal proof of ownership. This has led to some problems where the purchase procedure hasn't adapted to the new legal documents.

Using the electronic bill with the above work flow provides a vulnerability between steps 2 and 3. Step 2 can be generated even without the existence of a physical car. A malicious seller could generate multiple apparently original bills and give to multiple buyers before they realize the car doesn't exist physically. A certain level of *trust* must exist between buyer and seller for the exchange to happen, and this *trust* can be malignantly used. This fraud is specially common on car sales between private owners.

Due to the lack of immediate retaliation available to the criminal, there is a time frame where the fraud could be repeated multiple times, even if finally, the car exchange takes place with an innocent buyer.

B. Proposal

The model will focus in providing some network security characteristics such as privacy, authenticity and secrecy as a way to establish connectivity between a client to the blockchain through a Server Wallet. The context diagram in Figure 1 shows the general stages (circled numbers) a user must follow to get or set information in a vehicle:

[2]The introduction of the Advance Electronic Signature published by the Official Journal of the Federation via https://www.dof.gob.mx/nota_detalle_popup.php?codigo=5457756

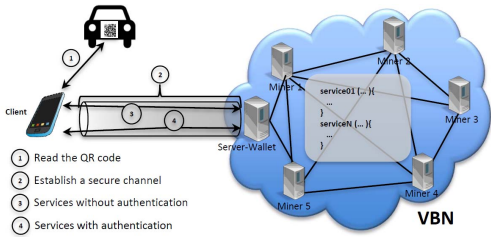

Fig. 1: Context Diagram from the client perspective.

1) Gets information of a vehicle through a QR code. This stage requires a previous step where the manufacturing company generates and inputs the initial information about the vehicle, after it has been manufactured. Section IV-B describes this procedure.
2) Establishes a secure channel between the user and the *Server wallet*. We have used the TLS protocol because it is standard in e-commerce and it has been subject to a lot of verification proofs. This stage is detailed in Section IV-C.
3) Requests a list of services such as provenance or traceability from *Vehicle Blockchain Network* (*VBN*). Some services will require authentication, and others won't. This stage is detailed in Section V.
4) Evaluates the role of the client (owner, government, manufacturers, etc) and fulfilling the requested transactions based on the evaluation. For example: the manufacturer requesting to build the original information for a vehicle inside the *VBN* (details in Section VI).

IV. Secure communication between agents

The section explains the notation used throughout the document; describes the participant agents in the model, the QR code generation and the role evaluation. Finally, it explains how the communication between the agents is securely carried out using the TLS protocol.

A. Notation

- **Message part:** $m_1; m_2$ denotes that message m_1 is concatenated with m_2; $[m_1 :: m]$ denotes a list of messages, being m_1 the first element. N_a, N_b, N_c denote Nonces, it is random unguessable numbers.
- **Cryptographic part:** two main mechanisms:
  - Symmetric cryptography, the same key is used to encode and decode a message; $\{\!|m|\!\}_{K_{AB}}$ denotes that message m has been ciphered under the shared key K_{AB} (known by agents A and B).[3]
  - Asymmetric cryptography, two keys are used to encode and decode messages; an agent (e.g. A) has a key pair (public $K_{pub(A)}$ and private $K_{priv(A)}$). One key used to encode and the other one to decode (reciprocally); usually, key $K_{priv(A)}$ is kept in secret.

[3]An agent is a computer process that can establish a network communication. Sometimes the terms agent or client will be used indistinctly.

- **Network part:** Table I describes network communication and security protocols. Each security protocol is illustrated in Alice and Bob notation, which is formed by the initial knowledge of the agents and a set of message steps. Every step consists in sending or receiving messages under a network context. Before sending a message, an agent can execute an operation (that might be dependent of the previous message), as part of a local process.

TABLE I: Conventions in the network level.

Abbreviation	Description
$A : ik$	Initially agent A knows ik.
$n.A \rightarrow B : m$	At step n agent A sends message m to agent B, which B receives.
Local process: $m2 = f(m)$ $n2.B \rightarrow A : m2$	B generates $m2$ (from $f(m)$) before sending it to A in step $n2$.
Broadcast: $A \rightarrow [B, C, D] : m$	A broadcasts message m to agents B, C and D.

B. Client-Wallet Participants and the QR Code

A QR code can be read quickly by many modern mobile devices. Its main objective is to scan the code from a non-digital media and obtaining some corresponding digital information in the device e.g., URLs, vCard, plain text, etc.

Our solution includes an application able to read a QR code (denoted as the client C), which connects with the *VBN* in order to do different operations. While requesting communication, C connects with a *Server wallet*, denoted as W, then, W connects with the set of the *VBN* denoted as $[\mathsf{M}_i, .., \mathsf{M}_n]$. The QR code must be generated by the manufacturer, who must place the code physically in some part of the vehicle. Currently, vehicles have a vehicle identification number (id) graven in some part, which must coincide with one of the attributes of the QR code information.

Let the QR code content the following data: *id=vehicle identification number*, *trademark*, *model*, *class*, *version*, *number of cylinders*, and among *others*. These attributes are the initial description of a vehicle, and cannot be changed over time. An example in JSON format is shown as follows:

```
{   id:          "1FMYU02Z97KA580G2",
    tradeMark:   "abcd",
    model:       "2012",
    class:       "auto",
    version:     "TA XLS 4X2 I4 TELA 4 CIL",
    cylinders:   "L4"
}   ::           others
```

The QR code information is formatted in plain-text because such an information is not sensitive and the goal is for anyone to read it at anytime. This information represents part of the content of the first transaction in a block of a vehicle.

C. Establishing a secure channel

Transport Layer Security (TLS) and its predecessor, Secure Socket Layer (SSL), are *cryptographic protocols* that work over the transport layer of the Internet Protocol (TCP/IP model) and has previously verified, [15], [16]. This protocol allows client/server agents to communicate securely by getting a session key K_{CW}, which let tunnelling a hostile network like Internet. This key will be used in the following transactions for encryption and decryption operations.

In the TLS protocol only the server is authenticated, but not vice versa. The session key generated between C and W is kept in secret, but W cannot be safe about C identity. Hence, an authentication step is required, which is detailed in Section V-B2.

TABLE II: Get Service Protocol.

(a) Authentication is not required.

Initial Knowledge
$C : C; QR; K_{CW}$
$W : W; K_{CW}$
1.-$C \rightarrow W : \{\!\|C; QR; n\|\!\}_{K_{CW}}$
Get a service $d = getService(QR, n)$
2.-$W \rightarrow C : \{\!\|C; d\|\!\}_{K_{CW}}$
Security Property Secrecy of n known by $[C, W]$

(b) Authentication required.

Initial Knowledge
$C : C; QR; K_{CW}; Y$
$W : W; K_{CW}; Y$
1.-$C \rightarrow W : \{\!\|C; QR; n; Y\|\!\}_{K_{CW}}$
Get a service $d = getService(QR, n)$
2.-$W \rightarrow C : \{\!\|C; d; Y\|\!\}_{K_{CW}}$
Security Property Secrecy of (n and Y) known by $[C, W]$

V. Get services

We will separate the specification of services by those who need authentication and those who don't.

A. Getting services without authentication

Some examples of services without authentication are: i) Verification of existence of current theft or debt reports. ii) Inquire of some basic characteristics such as *trademark*, *model*, *class*, *version* or *number of cylinders*. Table II(a) describes how to get these services:

- The initial knowledge is formed by all the values learned by the client, C, and the *Server wallet* W after completed the secure channel stage (described in Section IV-C).
- The next two steps are used to get services:
  1) The request of a service consists in sending an identity C, the data of the QR code, and a number n denoting the number of service that is requested to be consumed. C ciphers these messages using the secret key K_{CW} obtained in the secure channel stage.
  2) The *Server wallet* W (having access to the *VBN*) verifies if such a service can be consumed (according permissions) and then, returns the answer d; ciphered using, again, K_{CW}. Finally, C decodes the message and shows the information d to the user.
- In this request the service number n is the only data that is insured and is known only by the protocol participants.

B. Get Services with authentication

Sometimes it will be necessary to know who is consuming some services. For these cases, clients must be authenticated. Transport Layer Security (TLS) allows client/server agents to communicate across a hostile network, but eventually, only the server is authenticated. Mutual authentication requires a

public key infrastructure for clients, which we achieve by mean of a *registration process*.

1) Registration with the Server wallet W: In order to have access to the *VBN*, a client C links it through W, hence, this part consists in establishing a *username* and a *password* between them, which will be mapped with an identity. The *username* must be a valid email; and the *password* will be created by the user, and won't be sent through the network. Table III summarizes the protocol, described as follows:

- Firstly, C must establish a secure channel with W. In this step all exchanged messages are ciphered using the symmetric key K_{CW}, which was obtained during the TLS protocol. Hence, K_{CW} appears in the initial knowledge of the protocol.
- Secondly, C must generate his key pair (public and private) through his username and password executing the following steps:
  1) C sends a nonce, his username denoted with his email @, and a list of personal data Pd.
  2) After receiving, W sends back to C via an email server C_e a data obtained of the client's personal information, Pd and a challenge $X = f(Pd)$.
  3) Then, C must generate the final password P, but only the hash is sent, $Hash(P, X, N_C)$, together with the nonce of the first step like a freshness property. Sending such a hash, we can ensure that P will be only known by C.
  4) Having accepted the hash, W creates a public and private key for the client ($K_{pub(C)}$ and $K_{priv(C)}$), which is mapped with C and the hashed password; the *Server wallet* provides to the C the public and private key; which will be used to do transactions.
- We have verified this protocol using The AVISPA-SPAN Tool, available via http://www.avispa-project.org/ finding the secrecy property: the password P to be known only by C; $K_{pub(C)}$ and $K_{priv(C)}$ to be known only by W and C.

Note that, any user could transact a lot of accounts with W, but each of their vehicle-wallet would be empty.

TABLE III: Client registration process.

Initial Knowledge
$C : K_{CW}; [@ :: Pd]$
$W : K_{CW}$
1.-$C \rightarrow W : \{\!\|N_C; [@ :: Pd]\|\!\}_{K_{CW}}$
$X = f(Pd)$
2.-$W \rightarrow C_e : \{\!\|@; Pd; X\|\!\}_{K_{C_eW}}$
C generates password P
3.-$C \rightarrow W : \{\!\|@; Hash(P, X, N_C)\|\!\}_{K_{CW}}$
W generates a pair key ($K_{pub(C)}$ and $K_{priv(C)}$)
4.-$W \rightarrow C : \{\!\|C; K_{pub(C)}; K_{priv(C)}\|\!\}_{K_{CW}}$
Security Properties
Secrecy of P known by $[C]$
Secrecy of $K_{pub(C)}$ known by $[C, W]$
Secrecy of $K_{priv(C)}$ known by $[C, W]$

2) Client/Server wallet Authentication: The authentication process trusts the users' *username* (email) and *password*, the user is responsible for its secrets. However, it is well known that any secure system cannot trust only in a username and password for control access. It is important not only to encrypt the sensible information, but also to implement more security mechanisms (including authentication procedures) to avoid possible attacks.

The goal of this stage is to establish authentication between a client C and a *Server wallet* W after having established a secure tunneling. Note that the public and private key of the user (received in the registration process) are not relevant during the authentication process, and they are only used to execute smart contract transactions. Table IV describes the authentication part and the explanation is as follows:

- Both C and W store all knowledge accumulated from previous steps. At this point, they have established a secure tunneling using the TLS protocol and have obtained the register procedure. So, the initial knowledge of C consists in knowing the session key K_{CW}, his password P and the secret X. On the other hand, W knows the same session key, C public and private key, personal information of C, secret X and the hashed $Hash(P, X, N_C)$. The following steps are proposed:
  1) C sends challenge N_C to W. W can map @ with C and build $hash(@; C; W; N_C)$. This message is ciphered with K_{CW} because the session key has been destined for them.
  2) W responds to C with a new challenge that includes its name C (which is mapped with @) and a hashed message Y composed by N_C and X both messages must be known by C. Again, all ciphered with K_{CW}.
  3) Once C has accepted the previous step, it means that he has deduced all previous messages and he has authenticated to the W, so, C sends back his username @, the challenge response N_C as a freshness property and his password concatenated with P, X and N_C.
- After having verified the protocol with AVISPA-SPAN tool, we found a strong authentication property: the protocol provides *authentication* from C to W on message N_c and the inverse (W to C) on message @.

TABLE IV: Authentication Protocol.

Initial Knowledge
$C : P; X; K_{CW}$
$W : Hash(P, X); X; [@ :: Pd]; K_{CW}$
Authentication
1.-$C \rightarrow W : \{\!\|W; @; N_C; Hash(@; C; W; N_C)\|\!\}_{K_{CW}}$
2.-$W \rightarrow C : \{\!\|C; Hash(N_C, X)\|\!\}_{K_{CW}}$
$Y = hash(N_C, hash(P, X))$
3.-$C \rightarrow W : \{\!\|@; N_C; Y\|\!\}_{K_{CW}}$
Security Properties
C authenticates W on N_C
W authenticates C on Y
Secrecy of P known by $[C]$

After authentication, the client can do different basic operations like change password, close a session, edit its profile; and request services to the *VBN* (through the *Server wallet*) as explained below.

3) Get Services with authentication: Some examples of services available only after an authentication process are: i) a list of previous and current owner; ii) how many times the vehicle has been arrested; iii) the number of penalties; iv) a history of taxes.

Table II(b) describes the protocol about how to get services when authentication is required. Different from Table II(a), when the service does require authentication, the hashed message Y is included in steps 1 and 2. Here, after step 1, the W verifies if such a service can be consumed and then, returns the answer d ciphered. Both n and Y are kept in secret by the participants.

VI. Transactions

The transactions involve one or more atomic operations, which can be executed by any user as long as it has the adequate permissions (Subsection VI-A). The transactions are sent from the client, through the *Server wallet*, and executed in the *VBN* using a security protocol (Subsection VI-B).

A. Users and Roles

The users can hold one or more of the following defined roles:

- Any **User** will have enough permissions to interact with the system in order to obtain information, $getInfo()$, on the vehicle (Section V). This role is also able to function as a buyer, $buy()$, but to perform this action it must be authenticated.
- The **Manufacturer** is the only role capable of inserting new vehicles into the network. He is the one who creates, $registry()$, the initial transaction with the genesis information this action requires authorization given by the Government, $authorizeReg()$.
- Every car has one and only one **Owner** at a given time. The owner will be the only one that has permission to change ownership, $sell()$, but also to request the government to change its legal status, $reqChangeLegalStatus()$. In addition, he can request and authorize a Helper to add some information about a car, $reqAddInfo()$.
- The **Government** approves for a user to have the role of Manufacturer, $addM()$, and also have the ability to change some of the legal status of a vehicle when legal situations are involved, $changeLegalStatus()$. Each car have different legal status: *stolen*, *arrested*, *penalty*, *registered owner*, *plate*, *tax history*, etc.
- The role **Helper** is the one who receives a temporal permission from the Owner to set one transaction on the blockchain, $AddInfo()$. The time frame for the temporal permission will be authorized by the Owner of the car, $authorizeAddInfo()$, who will set a special transaction to give the helper the permission. The helper role will be used for transactions that change one or more data of the car's properties.

B. Set Transactions

As shown in Table V, the participants of a transaction are the Client C (who will have a role i.e. **O**wner, **M**anufacturer, etc) and the *Server wallet* W.

Initial knowledge: C knows the QR code through a scan process with a car; a secret shared key K_{CW} established with W in the secure channel process; a proof to have been authenticated with W, N_{CW}; and the transaction function $f()$ that C wishes to do. W knows also K_{CW} and N_{CW}.

1) C builds the operation data *op*, the signature of the operation $\{\!|op|\!\}_{K_{priv(C)}}$ and sends it to W; all ciphered with the session key K_{CW}. W verifies the session, with N_{CW}, accepts the previous message and proceeds to build the transaction t.
2) W sends the transaction to one miner, M_X (e.g.), which is within the *VBN* and acts like a parity node[4]. M_X develops block b', not mined yet.
3) M_X broadcasts the message to all miners $[\mathsf{M}_i..\mathsf{M}_n]$ in order to execute a mining process, $b = mine(b')$, within the *VBN*.
4) Miner M_X having (calculated or received) the mined block, b, returns it again to W.
5) Finally, W responds r to the client as a proof that the transaction was carried out successfully.

The security properties provided are: keeping in secret the operation *op*, the transaction t, the block b', and the mined b between the corresponding agents as shown in Table V. Note that during the verification, the broadcast request (step 3) was modified in the SPAN model and simulated as another trusted agent (acting as a miner).

TABLE V: Set Transactions Protocol, authentication required.

Initial Knowledge $C : C; Qr; K_{CW}; N_{CW}; f()$ $W : W; K_{CW}; N_{CW}$
Set a transaction $op = operation(Qr, f(), N_{CW})$ 1.-$C \rightarrow W : \left\{\!\!\left\{ C; op; \{\!\{op\}\!\}_{K_{priv(C)}} \right\}\!\!\right\}_{K_{CW}}$ $t = setTransaction(C; op; \{\!\{op\}\!\}_{K_{priv(C)}})$ 2.-$W \rightarrow \mathsf{M}_x : \{\!\{C; t; N_W\}\!\}_{K_{WM_x}}$ $b' = block(t)$ 3.-$\mathsf{M}_x \rightarrow [\mathsf{M}_i..\mathsf{M}_n] : \{\!\{b'; N_W\}\!\}_{K_{M_i..M_n}}$ $b = mine(b')$ 4.-$\mathsf{M}_x \rightarrow W : \{\!\{C; b\}\!\}_{K_W}$ $r = f(t)$ 5.-$W \rightarrow C : \{\!\{C; r; N_{CW}\}\!\}_{K_{CW}}$
Security Properties Secrecy of (op) known by $[C, W]$ Secrecy of (t, b' and b) known by $[W, \mathsf{M}_i, ..\mathsf{M}_n]$

VII. Verification and Proof of Concept

Verification: SPAN software (*S*ecurity *P*rotocol *AN*imator for AVISPA) was used to verify the protocols over a virtualized environment. The verification proof we have carried

[4]A parity node is a miner that is close with W and have usual peer to peer mutual communication.

out are those of Table II, III, IV and V as explained in the respective sections. In our proofs the intruder follows the Dolev and Yao Model, [17], where an intruder is not able to do cryptoanalisys but he can overhear, intercept, and synthesize any message.

Proof of concept: The feasibility of the operations is shown in the proof of concept. It does not focus on the computational security aspects. The software components installed to simulate the model are: i) Node Version 10.15, NPM Version 6.4.1, and Ganache CLI v6.4.3 to simulate the Blockchain Network based on Ethereum; ii) Metamask Version 6.6.1, to link the browser with the Blockchain, it simulates the *Server wallet*; and iii) Remix and web3.js to write Solidity contracts in the Blockchain, simulating the actions of the users. Figure 2 (left part) shows the smart contract of three operations ($addM()$, $registry()$ and $getInfo()$). The simulation was executed with three type of users: Government, Manufacturer and the Attacker; shown in the figure in the right part.

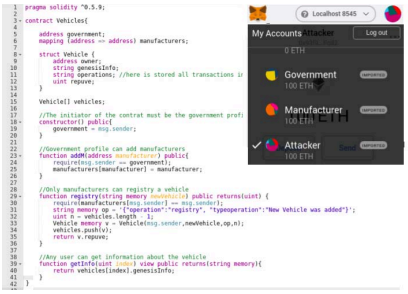

Fig. 2: Smart Contract example and type of users.

The Government can add manufacturers but cannot registry new vehicles; manufacturers can registry new vehicles, but cannot add other manufacturers; the attacker can get information, but cannot do anything else. Security access was validated by trying to execute operations without the adequate credentials. Note that the smart contract of Figure 2 is only one part of the global system.

VIII. Conclusion

We have proposed a distributed model to store historic transaction records about vehicles ownership through blockchain technology and security protocols. It involves the creation from an initial block until different distributed transactions such as selling, change of legal status, add information, etc. This information would become very valuable poured on public services and available for quick access, e.g., theft and debt status, different owners, etc.

The model includes the description of all participants and how they interact in terms of messages. It takes care security aspects in the exchange of messages, detailing all the security properties each of the protocols provide and how it is connected with the transactions of the blockchain.

Even though our model presents the transaction logic followed specifically on some Latin American countries, it can be applied to other countries around the world who are looking to have a digital ecosystem that have better control over the transaction history a vehicle can have throughout its life cycle. Hence, our model may contribute to extend the paperless culture and help solve some loopholes on vehicle purchase transactions that can be exploited to commit a fraud.

We conclude that Blockchain is a better electronic solution than a traditional database ledger; eventhough a database can keep a historical record of changes, blockchain simply adds new blocks when modifying data, hence having a natural historical log persistance. There is still a lot to do to have a fully functioning solution, and our ongoing work focuses on doing computational analysis concerning performance, communication overheads, and scalability.

References

[1] R. Chatterjee and R. Chatterjee, "An overview of the emerging technology: Blockchain," in *2017 3rd International Conference on Computational Intelligence and Networks (CINE)*, Oct 2017, pp. 126–127.

[2] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Sok: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Security and Privacy (SP), 2015 IEEE Symposium on*. IEEE, 2015, pp. 104–121.

[3] Q. DuPont, "Experiments in algorithmic governance: A history and ethnography of "the dao," a failed decentralized autonomous organization," in *Bitcoin and Beyond*. Routledge, 2017, pp. 157–177.

[4] H. Pagnia and F. C. Gärtner, "On the impossibility of fair exchange without a trusted third party," Technical Report TUD-BS-1999-02, Darmstadt University of Technology ..., Tech. Rep., 1999.

[5] B. Garbinato and I. Rickebusch, "Impossibility results on fair exchange," *10th International Conferenceon Innovative Internet Community Systems (I2CS)–Jubilee Edition 2010–*, 2010.

[6] J. Stark, "Making sense of blockchain smart contracts," *CoinDesk, Published on June*, vol. 4, 2016.

[7] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[8] A. Ramachandran and M. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," *CoRR*, vol. abs/1709.10000, 2017.

[9] M. McConaghy, G. McMullen, G. Parry, T. McConaghy, and D. Holtzman, "Visibility and digital art: Blockchain as an ownership layer on the internet," *Strategic Change*, vol. 26, no. 5, pp. 461–470, 2017.

[10] K. Poudel, A. B. Aryal, A. Pokhrel, and P. Upadhyaya, "Photograph ownership and authorization using blockchain," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, vol. 1. IEEE, 2019, pp. 1–5.

[11] R. DeMarinis, H. Uustalu, T. Fay, D. Paniscotti, T. Parvits, R. Rajkumar, and T. Junning, "Application framework using blockchain-based asset ownership," Nov. 16 2017, uS Patent App. 15/592,803.

[12] N. Baranwal Somy, K. Kannan, V. Arya, S. Hans, A. Singh, P. Lohia, and S. Mehta, "Ownership preserving ai market places using blockchain," in *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, pp. 156–165.

[13] L. Arkusha and N. Chipko, "Methods of committing and investigation of fraudulent motor vehicle transactions," *Russ. LJ*, vol. 6, p. 126, 2018.

[14] K. Geldenhuys, "Insurance fraud - the criminal mind behind vehicle crime," *Servamus Community-based Safety and Security Magazine*, vol. 111, no. 7, pp. 33–35, 2018.

[15] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, J. Pan, J. Protzenko, A. Rastogi, N. Swamy, S. Zanella-Béguelin, and J. K. Zinzindohoué, "Implementing and Proving the TLS 1.3 Record Layer," in *SP 2017 - 38th IEEE Symposium on Security and Privacy*, San Jose, United States, May 2017, pp. 463–482.

[16] B. Blanchet, "Composition theorems for cryptoverif and application to tls 1.3," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018, pp. 16–30.

[17] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, March 1983.