DeMetrA - Decentralized Metering with user Anonymity and layered privacy on Blockchain

Mario Vasile and Bogdan Groza
Faculty of Automatics and Computers,
Politehnica University of Timisoara, Romania
Email: vasile.mario22@gmail.com, bogdan.groza@aut.upt.ro

***Abstract*—Wear and tear are essential in establishing the market value of an asset. From shutter counters on DSLRs to odometers inside cars, specific counters, that encode the degree of wear, exist on most products. But malicious modification of the information that they report was always a concern. Our work explores a solution to this problem by using the blockchain technology, a layered encoding of product attributes and identity-based cryptography. Merging such technologies is essential since blockchains facilitate the construction of a distributed database that is resilient to adversarial modifications, while identity-based signatures set room for a more convenient way to check the correctness of the reported values based on the name of the product and pseudonym of the owner alone. Nonetheless, we reinforce security by using ownership cards deployed around NFC tokens. Since odometer fraud is still a major practical concern, we discuss a practical scenario centered on vehicles, but the framework can be easily extended to many other assets.**

***Keywords*—Decentralized Metering; Blockchain; Privacy**

I. Introduction and motivation

Putting trust on sellers from the second-hand market requires a reputation based system (as implemented on e-bay or related websites) which is not friendly for newcomers and may be in turn deceived. The wear and tear degree of a product, as suggested by specific metering sensors, is however a good indicator for its current state and most assets have at least a meter of some sort that provides such information. For example, digital single-lens reflex cameras, DSLRs have shutter counters, cars have mileage counters (odometers) and even gadgets such as some notebooks or smartphones count the recharge cycles of their batteries. Needless to say, all these counters may be subject to modifications either innocuous (e.g., shutter replacement) or adversarial in nature (e.g., changing miles on vehicles by altering the information stored on ECUs). A publicly auditable database that allows to check the history of a product may help in this respect. Of course, this is is not perfect since even new products may have flaws but it will give more confidence to potential customers and it will help in establishing an honest market value for the asset.

The goal of the application that we develop is to provide a decentralized system for storing some metering information for a product and to allow the owner to update this periodically (either manually or automatically). The history of the product can thus become publicly available. Due to the more representative market value we place cars and odometer values at the core of our work, but the system can be easily extended to other assets, e.g., smartphones, laptops, cameras, etc. that have a counter of some sort.

The second-hand vehicle market is affected by odometer fraud in a visible way. Second-hand cars represent the largest car sales in terms of both value and volume. The second-hand car market is much bigger than the market for new cars [3]. In 2012 alone the second-hand market for cars in Germany, Italy, France, Spain and UK accounted for 24 million of used cars while new cars toped at 9 million cars. Lowering the mileage of a vehicle can significantly increase its resale value. The problem becomes more relevant in countries were car resales are higher. Financial damage is infringed not only on the customer who pays much more than the value of the product, but also on the society since a car that is in worse shape than claimed requires more replacement parts that may add up to significant economic loses. Besides financial damage the phenomenon may impact the quality of life as well since heavily used vehicles may pollute more. The safety of the driver, passengers and other road participants may be affected as well since mechanical devices, e.g., cables or wires, may fail after prolonged use. Nonetheless, odometers modifications may affect seller credibility and the seller may be innocent victim to modifications that were done by previous owners, etc. So the addressed problem is realistic.

Non-profit initiatives for combating mileage modifications exist, for example Car-pass[1]. In a recent report, Car-pass claims costs in the order of billions for European citizens due to odometer fraud [1]. According to a new report from the European parliament [3] estimations vary from 10% and 50% on the number of second-hand traded cars that have modified odometers. These numbers show that the problem is widespread and represents a serious concern across the entire European Union. Notably, this happens despite the fact that most European countries have a legislation that is in favor of preventing mileage modification as the report in [2] concludes. It is not surprising that the second-hand car market has a low trust level when compared to other markets according to [3]. The study points to considerable differences between the Market Performance Indicators (MPI's) [3] for the top and the lowest ranking Member States respectively.

The system that we propose is a decentralized public database based on blockchains for storing different metering

[1] https://www.car-pass.be

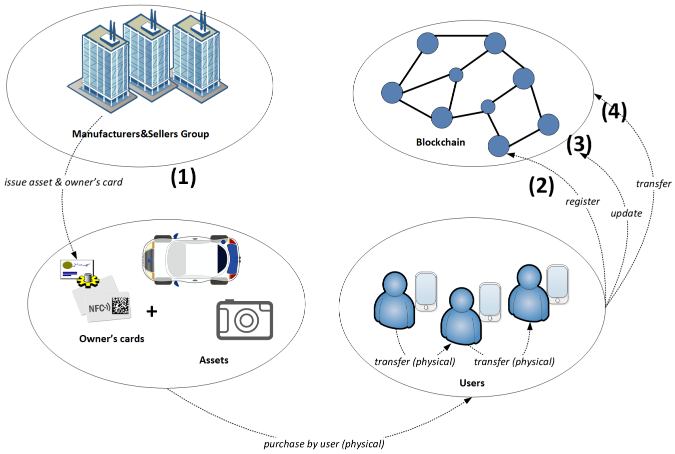

Fig. 1. DEMETRA: overview of the addressed setup

for an asset. A graphical depiction of the setup is in Figure 1. If we take cars for example, the odometer value is stored in the database and the potential buyer can verify the records for the odometer. When the item is first sold to the user, the entity who sells the asset will ask the client if he wants the product to be register on the blockchain. If the client approves, an owner card is issued in step (1) and the authorized seller will upload the information about the product on the blockchain, i.e., step (2) and will pass the credentials for that asset to the customer. The first data written by the authorized seller represents the information for identifying the item (it may include the serial number, manufacturer, model, year of fabrication and year of purchase, etc.). The data will also be signed by the seller with the help of identity-based signatures (IBS). The client will be able to update further the asset with metrics along the product lifetime, i.e., step (3). If he later wants to sell the product he can provide a history of it to the potential buyer. Once the asset was successfully sold, the previous owner will transfer the asset to the new owner, i.e., step (4). While the system aims to increase the trust level on the product it may be also merged with a reputation based system in case that product owners agree. We underline that the assets registered on the blockchain will not have a link to the current owner. The first registration for a product in the blockchain is done when the item is first sold by the vendor which signs the data with his id. The user will then have a random pseudonym and can sign further data for the object.

II. Protocol design

We now discuss our protocol design for DEMETRA. We start from some theoretical background, then we proceed to the protocol design goals and protocol description.

A. Foundations: blockchains and ID-based cryptography

In this section we first discuss the two theoretical concepts behind the solution: blockchain and id-based cryptography. We then discuss some challenges regarding blockchain storage and certificate revocation.

Blockchains. The Blockchain is an emergent technology and it has been recently proposed for protecting in-vehicle data, e.g., [7], [6], [4]. A blockchain is a list of records called blocks which can be appended for an unlimited number of times. Each block from the blockchain contains two parts: the hash of the previous block, and the current data and thus the hash of the block is computed over the data field plus the previous hash. This ensure that integrity can be traced back to the first block of the Blockchain. In this way the blocks are linked by a cryptographic hash, i.e., a cryptographic one-way function that is infeasible to invert. Since it is practically infeasible to invert the hash function or to find collisions in it, once a block is recorded in the chain the input can no longer be modified. If a block is modified then all of the following blocks will change and this will be detected by the network. Decentralized consensus can be further achieved in case that conflicts appear.

Incentives for storing the data. Because the system is decentralized, peers are required to store the blockchain on their own. If the system is used at a large scale, the blockchain size can increase significantly and thus incentives need to be provided to the keepers. The value for storing the blockchain comes from the public data which every owner of the asset will share with the network. For maintaining car odometers, since these are a problem at the European level, both governments and producers may share interest in maintaining the blockchain. For other products, on-line markets (such as ebay.com or olx.ro) may have interest in maintaining the blockchain since resellers can get a better price for the product and thus gain more interest in using their platform. Nonetheless, in case that more product history is available to the keepers, other actors such as insurance companies or sites that sell similar products may share interest in maintaining the blockchain infrastructure.

Achieving consensus. Establishing consensus between nodes is fundamental for maintaining the blockchain. Fortunately the consensus protocol is transparent and comes directly from the blockchain provider that we use, i.e., BigchainDb. To achieve consensus Bigchain uses Tendermint [10], a consensus protocol for blockchain that does not require mining. This is important since mining requires significant amount of computation which translates to electrical energy and costs while BigChain incurs no additional costs. Tendermint is able to establish consensus even if up to 1/3 of machines fail.

ID-based cryptography. Digital signatures are fundamental for proving the source for a piece of information. They provide non-repudiation since they can be verified by any neutral third party. In the most common applications, e.g. SSL/TLS, each user (or server) has a public-private key pair that is randomly generated. The user identity is linked to its private key in a public-key certificate that is in turn signed by a trusted third party. A certificate authority (CA) can map the identity to a public key by using a certificate but for this a public-key infrastructure (PKI) needs to be put in place. However, in the real-world, users are not identified by randomly generated keys but by a names, a serial number or some element that generally is not random. The identity-based setting was proposed by

Shamir in 1984 [11] and in this setting a user public key is his identity from the real world while the private key is generated by a trusted key generation center (KGC) with the use of a private master key. The public key could be any string that identifies a person, a product, it can be a serial number, an IP address, the name of the company or of an individual. This type of signatures eliminates some of the inconveniences related to maintaining a PKI. Since currently there is no id-based signature scheme available by default on popular frameworks such as .NET or Java JDK, we can rely on the original scheme from Shamir or the Guillou-Quisquater scheme [8], [9] as proof-of-concept. These schemes are easy to implement in any language that has support for arbitrary precision integers, e.g., BigInteger. As future development we consider embedding more advanced schemes such as pairing-based alternatives which also allow for identity-based encryptions [5].

ID based cryptography has a security shortcoming in the need for a trusted key generation center (which generates both the private and public keys and thus can use them instead of the genuine user) but limitations exist in the traditional CA setup as well. For example a corrupted CA may also issue duplicate certificates. But, in the CA setup the CA may only have access to the public-keys that are signed and thus cannot decrypt/sign information as it is not in possession of the private counterpart. Since our application relies on digital signatures rather than on public-key encryption, even in the case of a corrupted KGC the confidentiality of the user will not be broken and moreover, past authentic data from the blockchain can still be retrieved. Thus the limitations of ID-based signature are not of significant concern to our setup.

Certificate revocation. Certificate revocation causes more concerns for ID-based signatures than for conventional signatures. This is because a conventional public-key certificate includes a validity period but a name will hold as a valid identifier without an expiry period. To address this shortcoming a public revocation list must be maintained which contains all of the revoked identifiers. This is similar to conventional CA based solution which also require the presence of a certificate revocation list (CRL).

B. Design goals

We briefly enumerate the design goals of the proposed system as follows:

1) *Traceability.* The history of an asset, represented by a specific counter directly associated to the degree of wear for the product, e.g., shutter count, tachometer, etc., must be easy to trace. It remains the decision of the owner to also publish or not other informations about the product (attributes) in a layered disclosure scheme as we later discuss.
2) *Owner anonymity.* The owner remains anonymous, it is solely identifiable by a random public key under which he can store a single asset. More assets can be stored under a single public key, but this should be done cautiously since an owner may be identified by a particular asset, e.g., the possession o a not so common car.
3) *Layered disclosure.* Disclosure of product attributes is done via a layered approach by which the owner may decide to share only a specific layer of product information. The owner may choose to structure the information as he wishes and the information will be chained by the use of Merkle trees. The blockchain will store only the root node of the Merkle tree which is the hash of the previous nodes and contains the minimum amount of information.
4) *Easy to implement.* The solution must be based on down-to-earth cryptography and on support from existing libraries. We based our solution on simple existing cryptographic building blocks, e.g., SHA256, HMAC, AES, and on support for blockchain like structures offered by BigChain.
5) *Enhanced security by NFC tokens.* The solution must have an additional secure layer for the case when the environment from which the user publishes information, e.g., his smart-phone or notebook, is lost or compromised. For this purpose we store product secret keys on what we call the *owner's card.* This card is an NFC token which is used thorough the protocol steps. Once the user disconnects it from the phone, the phone stores only encrypted data about the product. Indeed, if the phone is fully controlled by an adversary, then the data can be compromised. However, since the data is stored in the blockchain, the adversary cannot modify already committed data since this can be easily detected by checking transactions in the blockchain. The best that the adversary could do is to compromise the privacy of the user by decrypting the locally stored data. The NFC token should be also printed with a datamatrix on it which contains the serial number of the product or some identifier.

Figure 2 suggests how we structure information for assets belonging to a user. The user (owner) is represented by the public key which is a random value. While the user has a user name this is only for convenience and may be selected as some random pseudonym. Since each user is uniquely identified by his public key, users sharing the same user-name may exists. Subsequenlty the assets are also uniquely identified by a public key. They also encapsulate more data such as the serial number and a name for the asset. The name of the asset is again chosen by the owner but this time it should represent the name of the product that he owns. Additionally the assets store the MAC of more product information which is structured as a Merkle tree. This allows the user to disclose only layers at his choice. The first layer also encapsulates a random salt value which ensures that subsequent blocks look random. The value of the data from each layer is stored encrypted on the mobile phone. The encryption keys are further stored on the NFC token which is the owner's card.

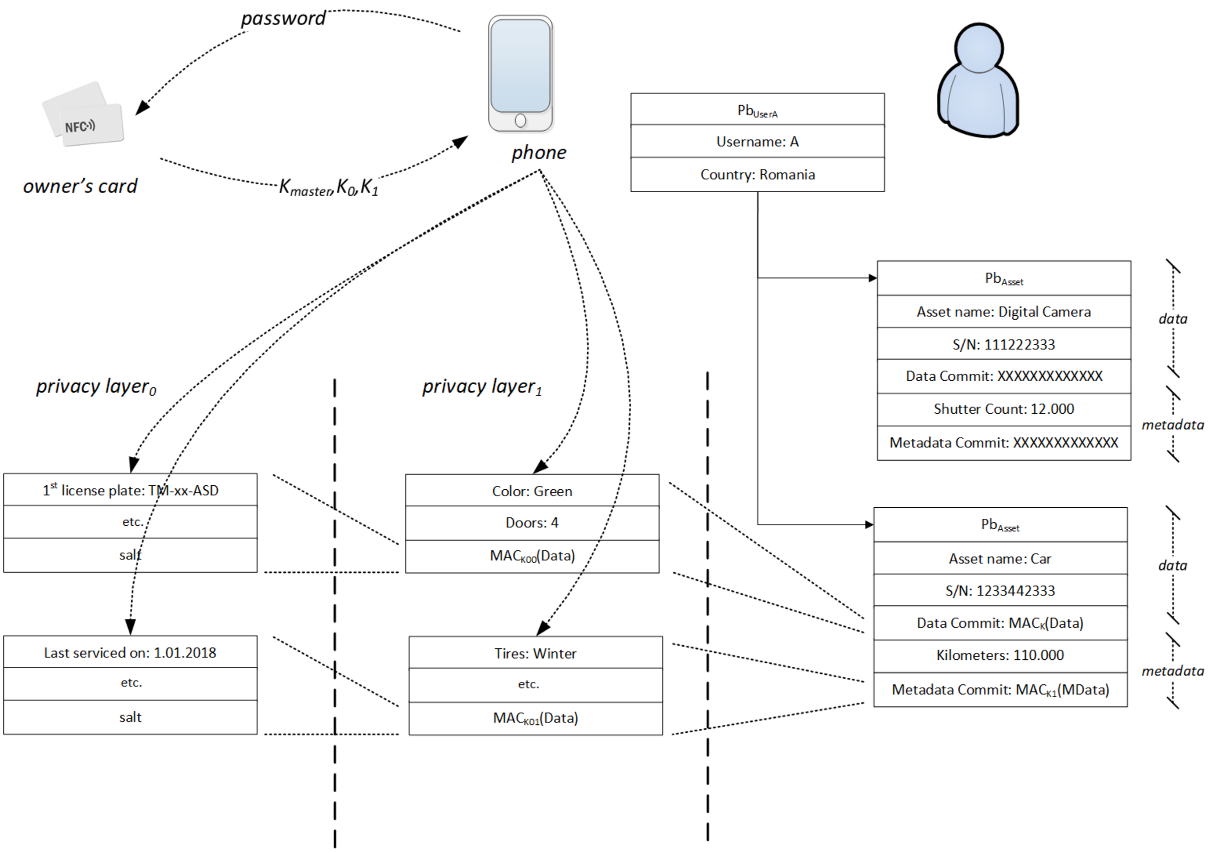

Fig. 2. Information structure for assets belonging to a user

C. Protocol description

We now give more details on the proposed protocol by discussing each of the protocol stages. For owner's card issuing, NFC card setup and the registration stage, the steps of the protocol are synthesized in Figure 3.

First, the manufacturer releases the owner's card which embeds a master key that will remain secret and stored only on the card. To setup the card, the owner via a phone application will select the number of layers for information disclosure λ and generate some random value $salt'$. To avoid confusing the users, a recommended value for λ and one data template should be available for each product. Giving details on this is out of scope for this work. The encryption and authentication keys key are derived from random material $salt'$ supplied by the user via a key derivation process which ensures that the manufacturer cannot guess the values of this key (despite the fact that the master key is also part of the derivation process). The encryption key and authentication keys are returned to the phone and will be used to encrypt and tag the values that are locally stored on the phone (this is to reinforce security on the smartphone app). The encryption and authentication keys are stored in the NFC card and will not be stored on the phone. In order to register the product, the user application will generate a second random value $salt''$ and set or retrieve product attributes $attr_i, i = 1..\lambda$. Each layer may have more than a single attribute. The attributes are hashed and the hash is sent to the NFC token which will encrypt and preserve the most recent value of the attributes for redundancy. The owner will further sign the data and sends this as a transaction to the blockchain.

III. Proof-of-concept implementation

Our implementation is focused around deploying the data in a blockchain and on evaluating some of the functionalities NFC (the development of a complete application is out-of-scope for this short communication).

A. System overview

The proposed system has 3 main components/actions: i) the registration of the asset which takes place when the asset is purchased for the first time and which is done by an authorized seller, ii) the update of the asset attributes which can be done by the owner whenever he considers to do so (in case when the car plays the role of an asset, the attributes is the mileage information as retrieved from an OBD tool via Bluetooth connectivity, an OBD gadget costs in the order of a few dozen euros), iii) transferring the asset from one owner to another when the asset is sold on the second-hand market. The first two procedures are outlined in Figure 2. To update the attributes of the asset the application will retrieve an identity-based private key as received from authorized seller (for convenience, we

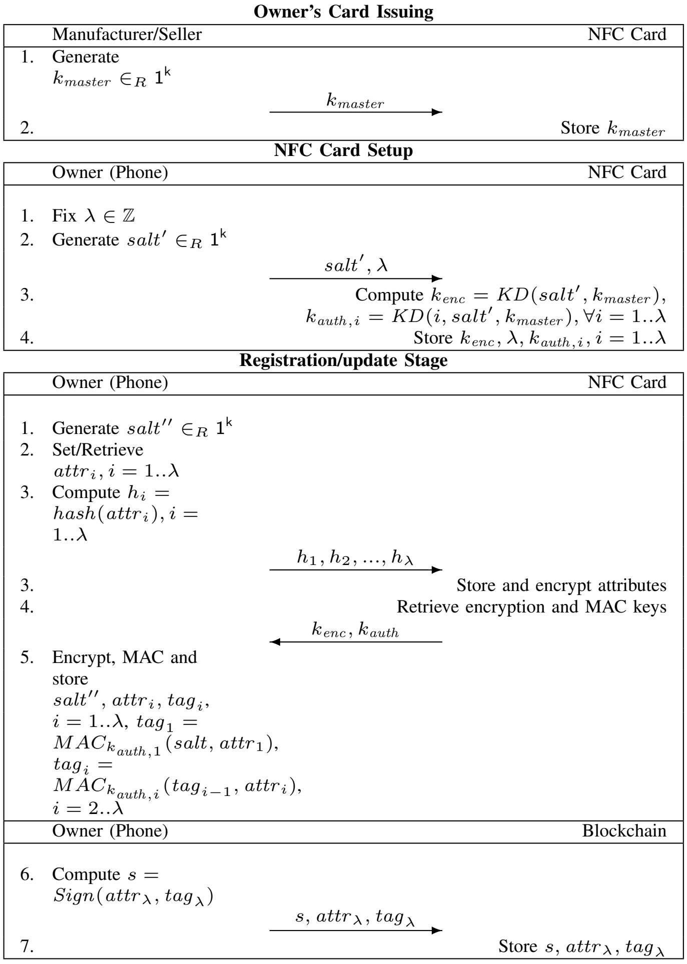
Owner's Card Issuing
Manufacturer/Seller
NFC Card
1. Generate $k_{master} \in_R 1^k$
k_{master}
2. Store k_{master}
NFC Card Setup
Owner (Phone)
NFC Card
1. Fix $\lambda \in \mathbb{Z}$
2. Generate $salt' \in_R 1^k$
$salt', \lambda$
3. Compute $k_{enc} = KD(salt', k_{master})$, $k_{auth,i} = KD(i, salt', k_{master}), \forall i = 1..\lambda$
4. Store $k_{enc}, \lambda, k_{auth,i}, i = 1..\lambda$
Registration/update Stage
Owner (Phone)
NFC Card
1. Generate $salt'' \in_R 1^k$
2. Set/Retrieve $attr_i, i = 1..\lambda$
3. Compute $h_i = hash(attr_i), i = 1..\lambda$
$h_1, h_2, ..., h_\lambda$
3. Store and encrypt attributes
4. Retrieve encryption and MAC keys
k_{enc}, k_{auth}
5. Encrypt, MAC and store $salt'', attr_i, tag_i$, $i = 1..\lambda$, $tag_1 = MAC_{k_{auth,1}}(salt, attr_1)$, $tag_i = MAC_{k_{auth,i}}(tag_{i-1}, attr_i)$, $i = 2..\lambda$
Owner (Phone)
Blockchain
6. Compute $s = Sign(attr_\lambda, tag_\lambda)$
$s, attr_\lambda, tag_\lambda$
7. Store $s, attr_\lambda, tag_\lambda$

Fig. 3. DEMETRA: Protocol overview

assume this is retrieved from a data-matrix diagram and is retrieved in a secure environment).

B. NFC token implementation

To test functionalities on NFC cards, we choose the Mifare DesFire EV1 which is an NFC card based on open global standards for air interface and cryptography. It is compliant with all 4 levels of ISO/IEC 14443A and use optional ISO/IEC 7816-4 commands. This type of card is ideal to be used to secure systems like access management, public transportation or closed-loop e-payment. The main feature of the card is his security.

The Mifare DesFire has an on-chip backup management system which can hold up to 28 different applications (like you would have 28 different cards) and every application can store up to 32 files. An application can be seen as a directory where a user can store data. An application will contain keys and files. Files will store data and keys will be used to limit the access for the files. Every application has a unique identifier named AID and is consist by a 3 byte array. The Mifare DesFire EV1 is available in 3 different sizes: 2kb, 4kb and 8kb and has a retention of the data of 10 years and the write endurance of 500 000 cycles.

TABLE I
EVALUATION OF READ OPERATIONS ON THE NFC CARD

Read data	512 bit	1024 bit	2048 bit
DES	100 ms	123 ms	125 ms
DES 2 keys	91 ms	119 ms	151 ms
DES 3 keys	104 ms	124 ms	165 ms
AES	94 ms	125 ms	182 ms

TABLE II
EVALUATION OF OPERATIONS ON THE NFC CARDS

Write data	512 bit	1024 bit	2048 bit
DES	201 ms	233 ms	282 ms
DES 2 keys	182 ms	251 ms	281 ms
DES 3 keys	210 ms	260 ms	290 ms
AES	192 ms	225 ms	282 ms

From the security point of view, the card has a unique 7 bytes serial number, has an optional RANDOM id, provides mutual authentication compliant with ISO/IEC 7816-4. Every card has one master key and up to 14 different keys for every application (up to 28 application per card). Supports hardware encryption using DES 56/112/168 bit keys and AES 128-bit key. The transmission channel will also be encrypted. Authentication is available at application level and has hardware exception sensors. The input frequency is 13.56 MHz. Because the card is compliant with the ISO/IEC 14443, the card doesn't require a battery. When a card is positioned in the proximity of the PCD antenna, the data is send through the high-speed RF communication interface.

We tested the speed of encryption and decryption for different input sizes using an Android phone and the SDK provided by Mifare to interact with the card. Taplinx is the name of the Java SDK provided by Mifare and which we used for the Android in order to interact with the Mifare DesFire Ev1. The phone used in the tests is Samsung Note 4 (N9100) and the card is a Mifare DesFire Ev1 2kb. Every value from the table represent the average of 200 cycles read or writes and include the authentication for the card. The results are shown in Tables I and II.

C. Blockchain implementation details

To implement a proof-of-concept application we used BigchainDb [2], an open source database that implements the blockchain technology. BigchainDb supports asset creation and transfer using a public-private key system. In order to create or update an asset the owner uses his private key for signing the transaction. The database is decentralized, meaning that multiple servers can run it independently and store the data. Each of the servers needs to maintain the same version of the data and to achieve this they run the aforementioned

[2] https://www.bigchaindb.com/

consensus agreement protocol for every block of data written in the blockchain.

As a proof of concept we created two applications: a REST api written in python for registering the asset to the network and an Android app through which the owner of the asset can update the item and transfer it to another person. The REST API will be used by the Authorized Seller for registering the asset to the blockchain. After the registration, the seller will pass a QR code containing the identifier of the asset, the private key and a transaction id to the buyer of the asset. The buyer of the asset scans the QR code from the invoice (or other document received when aquiring the product) with an Android app. He can choose to store the public and private key on the phone or some external device, e.g., an SD-card or an NFC token which can be kept in a safe place. The private key will be used only when updating the metering for the asset or to transfer owner rights. From the app he can view the entire history of his asset. The app allows also to create pair of public and private key and display each of them as QR codes. This will enable the transfer of an asset to another person by scanning the QR code of the public key from the customer phone. The app will allow displaying the history of an asset by scanning the QR code (public key of the asset). Two types of information are defined and stored in the system for an asset: Data and Metadata. Every transaction store information in a JSON format. The Data type holds informations which identify the product. An example of the data field is shown in Figure 3.

This data will be registered by the vendor when the item is first purchased. The originator signature field is an identity-based signature of the vendor and provides authenticity for the product on the network in case other items will be registered with the same information. The data field cannot be updated after it's creation even if you have the private key.

The Metadata field is more flexible and let user to customize the entries. It is a key-value dictionary with properties defined by the owner. We enforce with as a condition for validating the transaction that this field contains a specific property named MeterValue which holds the metering information of the asset (for car will be the odometer value, for phones can be the battery cycle count, etc). The metadata could contain any other information related to the product like the name of the current owner, the value paid for it, current country where the item is located. For anonymity reasons, in our application the owner name (part of an optional field) can be replaced by a random ID or even be leaved blank. An example of the Metadata field can be seen in the Figure 4. The system will provide a searching functionality for both types of information: data and metadata. Through the Android app the user will be able to check the history of other assets or update the metering information for his own assets.

IV. Conclusion

The system that we address provides an efficient way to keep the history of a product by relying on two modern concepts from cryptography: blockchains and id-based signatures. Users can trust that the history of the product remains unchanged on the blockchain based on strong cryptographic guarantees. Moreover, privacy is assured by a layered disclosure of the product attributes. The history of the product will increase the trust level and potentially give the owner a better price when he decides to sell it. We believe that such a system has value for both the owner of the asset and for the holder of the database. We are aware that the solution provided will not fully solve the trust issue since dishonest owners may still find a way to fill false information about the asset on the blockchain (e.g., cracking the counter report) and a reputation system may help in this respect. Due to inherent space constraints, this small research communication is restricted to the problem statement and some details on our proof-of-concept implementation. As current and future work we pursue the development of specific protocols and procedures for each step and hope to further extend this work.

Acknowledgement. This work was supported by a grant of Ministry of Research and Innovation, CNCS-UEFISCDI, project number PN-III-P1-1.1-TE-2016-1317, within PNCDI III (2018-2020).

References

[1] European Comission, Consumer Markets Scoreboard Making markets work for consumers 10th ed. https://fil.forbrukerradet.no/wp-content/uploads/2015/10/Market-Scoreboard-rapporten.pdf, 2015.

[2] nextcontinent: automotive finance study the european market and its future. http://www.nextcontinent.net/publications/automotive-finance-study-2016/download, 2016. Accessed: 2018-06-07.

[3] Research for TRAN committee, Odometer tampering: measures to prevent it. http://www.europarl.europa.eu/RegData/etudes/STUD/2017/602012/IPOL_STU(2017)602012_EN.pdf, 2017. Accessed: 2018-06-07.

[4] M. S. U. Alam, S. Iqbal, M. Zulkernine, and C. Liem. Securing vehicle ecu communications and stored data. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[5] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 2004.

[6] M. Cebe, E. Erdin, K. Akkaya, H. Aksu, and S. Uluagac. Block4forensic: An integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Communications Magazine*, 56(10):50–57, 2018.

[7] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak. Blockchain: A distributed solution to automotive security and privacy. *IEEE Communications Magazine*, 55(12):119–125, 2017.

[8] L. C. Guillou and J.-J. Quisquater. A "paradoxical" identity-based signature scheme resulting from zero-knowledge. In *Proceedings on Advances in cryptology*, pages 216–231. Springer-Verlag, 1990.

[9] L. C. Guillou, M. Ugon, and J.-J. Quisquater. Cryptographic authentication protocols for smart cards. *Computer Networks*, 36(4):437–451, 2001.

[10] J. Kwon. Tendermint: Consensus without mining. *URL https://tendermint. com/static/docs/tendermint. pdf*, 2014.

[11] A. Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer, 1984.