Augmented chain of ownership : configuring IoT devices with the help of the blockchain

Sophie Dramé-Maigné, Maryline Laurent, Laurent Castillo, Hervé Ganem

Augmented chain of ownership : Configuring IoT devices with the help of the blockchain

Sophie Dramé-Maigné[1,2], Maryline Laurent[2], and Laurent Castillo[1]

[1] Gemalto SA, 6 rue de la Verrerie, 92190, Meudon, France
[2] SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, 9 rue Charles Fourier, 91011, Evry, France

Abstract. The blockchain, distributed and unalterable ledger, has lowered the cost of ownership records. This mechanism that required the involvement of notaries and was once reserved for expensive items can now be used for every type of assets. We propose to use it to keep track of the ownership of IoT devices. By registering devices and their transfer into the blockchain, we build a chain of ownership that can be used to guarantee ownership before a sell or to track and warn owners of security threats faced by their devices. Two extensions are proposed. The first one facilitates remote configuration and key management when a single owner must configure a great number of devices. The second one offers potential users information about an IoT device. That information can be used to create a discovery service, inform access decisions or keep track of a device's state.

1 Introduction

In 2008, Nakamoto[17] introduced the concept of the blockchain, a public shared unforgeable ledger allowing participants to register transactions in a persistent and decentralized manner. If it was intimately linked with cryptocurrencies at first, the blockchain has since been used in other applications such as voting[4], online games[3], ride sharing[1], and many others[3]. With properties such as desintermediation, unforgeability, and decentralization, the blockchain is also very attractive to the Internet of Things (IoT). Its usability in such setting has been studied with mixed results [8,9,15,14].

Another common blockchain use case is the tracking of ownership of assets such as houses, cars or artwork [22]. Such initiatives have the power to combat corruption in countries where acquiring a land can be discussed with corrupt officials instead of the actual owner. Countries such as Sweden[4], Georgia[5], and

[3] https://gomedici.com/30-non-financial-use-cases-of-blockchain-technology-infographic/, Last checked Feb, 16th 2018
[4] https://www.reuters.com/article/us-sweden-blockchain/sweden-tests-blockchain-technology-for-land-registry-idUSKCN0Z22KV, Last checked Feb, 16th 2018
[5] https://cointelegraph.com/news/georgia-becomes-first-country-to-register-property-on-blockchain, Last checked Feb, 16th 2018

Ukraine[6] are each at different stages of implementing a land title project using the blockchain.

In this paper, we propose to use the blockchain to track IoT devices' ownership. The blockchain is a cheaper alternative to ownership records when compared to traditional methods that involve an outside authority such as notaries. It is also a simpler and faster process. Thanks to the decreased cost and added usability, ownership records can then be used for more humble assets such as IoT devices.

This mechanism can also be used to exchange device-related secrets, enabling remote configuration and efficient key management. IoT use cases can involve many devices deployed in various physical locations. This makes manual configuration inefficient. Smart grids are a good example of hundreds of devices that need to be deployed to cover the entirety of electricity grids. The deployment speed is highly impacted by the configuration method, as many devices need to be configured at once. By leveraging the chain of ownership published in the blockchain, we propose to facilitate remote configuration. Additionally, owners can use the same mechanism to efficiently manage the multiple keys used to remotely manage their devices.

The mechanism used to register device ownership can also be used to publish information related to IoT devices, either for safekeeping or for advertizing their characteristics to potential users. IoT use cases depart from classical ones that involve only a small number of known actors. The list of clients for one IoT device can be dynamic. In order for that device to be trusted with a task, potential clients may require some guarantees. For instance, a user could demand that the device be running the latest version of a software, hence ensuring that security patches have been applied, or simply that its list of communication protocols intersect with her own. Such information can be published on the blockchain.

Related Work Ownership tracking via a blockchain has already been implemented. On the Bitcoin blockchain, Colored Coins [22] can be used to track asset exchanges. On the Ethereum blockchain [7,24], smart contracts [23] can be programmed to do similar things. Other blockchains such as NXT [6] provide a native asset exchange. There are also front-end applications [5,2] that bridge several blockchains together to facilitate interoperation. These implementations are not IoT-specific but their general-purpose tokens are IoT-compatible. They are however only focussed on ownership record and cannot be used for key management or to share configurations.

In the academic literature, the transfer of ownership is addressed at the device level [18,20]. Ownership transfer is defined [21] as "the capability to pass ownership of a tag to a third party without compromising backward untraceability for the said party or forward untraceability for the previous owner." The focus is on key management and domain boundaries. The devices that are con-

[6] https://www.bloomberg.com/news/articles/2017-10-03/ukraine-turns-to-blockchain-to-boost-land-ownership-transparency, Last checked Feb, 16th 2018

cerned by these protocols are RFID tags. No record is kept of past owners. This article presicely focuses on these ownership records.

Symbol	Description
D	Device
$\{D_i\}_{0 \leq i < n}$	Family of n devices
id_i	Identifier of device D_i
O	Device owner
M	Device manufacturer
R	Retailer
$addr_A$	Blockchain address of A. $addr_A = Hash(pub_A)$
$(pub_A, priv_A)$	Public/private blockchain key pair linked to $addr_A$
s_i	Secret linked to device D_i
K_A	Master key of A
$k_{A,i}$	Symmetric key derived from K_A and id_i
tx_k	k^{th} blockchain transaction
out_j^k	j^{th} output of tx_k
$Prop_D$	Set of dynamic properties of device D

Table 1. Our notations

Organization Security assumptions and threat models are presented in Section 2. Based on the notations of Table 1, Section 3 introduces our tracking of ownership using the blockchain. Section 4 proposes an extension of the approach to configure IoT devices and manage keys for the sake of the owner. Section 5 turns to the users and explores what they can gain from it.

2 Security Considerations

2.1 Security assumptions

We operate under the following assumptions:

*A*1 *Secured blockchain keys*: Blockchain keys cannot be stolen, lost or otherwise compromised. This implies good key management.

*A*2 *Solid cryptographic primitives*: Our proposal uses cryptographic primitives such as signatures, hashes, or encryption. We assume these primitives cannot be broken.

*A*3 *Blockchain consistency*: Fundamental blockchain properties include consistency amongst nodes and consistency over time [19]. This implies that all nodes in the network will agree on blockchain history, the few last blocks excluded, and that accepted transactions cannot be modified. We assume these properties are verified and the blockchain history cannot be altered.

*A*4 *Blockchain capability*: All actors (M, R, O, and U) own a blockchain address, the corresponding public and private key pair, and the means of submitting or retrieving a transaction to or from the blockchain.

2.2 Threat model

Across our three proposals, we consider three types of attackers : a malicious new owner, a malicious previous owner, and a malicious uninvolved third party. We detail 8 possible threats involving these actors. These threats are summarized in Table 2.

Malicious previous owners This attacker's goal is to either fool a potential buyer by not providing the device after the sell has been concluded or to retain access to said device and gain access to sensitive data belonging to the new owner. As the previous owner, the attacker is in possession of the credentials that, at the time of the sell, allow access to the device. She also has the possibility to provision anything unto the device and is also able to produce a proof of ownership. When a device is sold, the previous owner can use her knowledge to gain access to sensitive information and maybe use the device as an entry point into the new owner's network. This defines Threat $T1$. A prospective owner can also be fooled by the previous owner and buy a device that will not be delivered. This defines Threat $T2$.

Malicious new owners The goal of this attacker is to gain access to sensitive information without authorization. As the new owner, the attacker has full access and full control over the device. After the sell, if the device has not been wiped clean, the new owner can extract potentially sensitive information related to the former owner from the device itself. This defines Threat $T3$. The new owner can also use the device's identity to gain access to previous owner's data. This can be done by interacting with users or devices that still recognize the device as being owned by the previous owner. This defines Threat $T4$.

Malicious third party This attacker goal is to pass as a legitimate device owner to fool a potential buyer, steal and re-sale a device, disturb the sell or gain information about the parties involved in the ownership change. When a public blockchain is used, the attacker has access to all information that transits through the blockchain. She can also produce and submit valid blockchain transactions. First, the attacker can try to clog the blockchain network. In this event, the miners would not be able to process the transaction signaling the ownership change. This defines Threat $T5$. Second, when the transfer occurs, the attacker may try to gain knowledge about the involved parties. This defined Threat $T6$. Third, the attacker may pretend to be the owner of a device she does not possess. This is Threat $T7$. Forth, the attacker may try to steal and resale the device. This threat is similar to Thread $T7$. Fifth, the attacker may fabricate a blockchain trace for a device that does not actually exist. This is Threat $T8$.

3 Asset ownership

3.1 Motivation

As previously mentioned, asset tracking is one of the most straightforward blockchain application. Assets that have been considered for this use case tend

Nbr	Attacker Type	Description
$T1$	Prev. Owner	Previous owner retains access to the device
$T2$	Prev. Owner	Proof of Ownership is produced but device is not provided
$T3$	New Owner	New owner extracts sensitive data from the device
$T4$	New Owner	New owner uses device to gain access to sensitive data
$T5$	Third Party	Ownership transfer cannot be completed
$T6$	Third Party	Attacker accesses sensitive information by eavesdropping
$T7$	Third Party	Attacker successfully masquerades as the device owner
$T8$	Third Party	Ownership chain with no corresponding device

Table 2. Threats

to be expensive and relatively immutable (i.e. land, cars, houses, paintings, etc). These objects' ownership will most likely already be tracked using third parties such as notaries or other government-sanctioned entities. The corresponding administrative procedures can be long and costly. By using the blockchain instead, trust in these third parties is no longer required. But that is not the only benefit. The cost of a transaction is also highly reduced. The transfer of ownership is a simple and quick operation. Ownership records are more likely to be up to date. For these reasons, ownership records do not have to be confined to expensive items. We propose to apply this principle to IoT devices.

Benefits of the proposal Keeping ownership records on the blockchain offers the following benefits:

- *Desintermediation*: Traditionally, changes in ownership must be attested and assisted by third parties. As the blockchain keeps a public proof of the transaction, they are no longer necessary.
- *Shared architecture*: When using a blockchain, one can take advantage of the architecture deployed by others. This use case does not require server deployment or federation of systems.
- *Decentralized storage*: This speaks to one of the fundamental blockchain property, persistence. Blockchain transactions will be stored in a decentralized fashion, protecting ownership record from loss and modifications.
- *Simplicity*: The process by which the ownership is transferred requires a single transaction. Its simplicity makes it highly usable, even to private individuals.
- *Lower costs*: Ownership transfer usually involves a third party. This third party will take a commission on the sale. The desintermediation therefore has the added benefit of lowering costs.
- *Traceability*: Ownership of an object can be traced back to its original owner. This gives a new buyer information about the life of a device, its age, maybe what it was previously used for, etc.
- *Proof of ownership*: Before buying a device second hand, the buyer can require a proof of ownership. He sends a message over to the vendor. The vendor must then provide a valid signature. The private key used to sign

the message must match the blockchain address that officially owns the device. This ensures that the device has not been stolen and that the vendor is allowed to proceed with the sale.
- *Security alerts*: When an incident affecting a big number of devices occurs, it is currently hard to track owners and warn them of the issue. Owners can be private individuals. They are not likely to follow best security practices. For that reason, in the event of a large scale IoT attack, being able to track and warn device owners could prevent further damage.
- *Availability*: By its distributed nature, the blockchain offers availability guarantees that servers cannot match.
- *Pseudonymity*: Traditionally, ownership records are nominative. This is natural as the proof of ownership is linked to one's identity. When the blockchain is used, such a proof is linked to what one owns, the correct blockchain private key. This enables pseudonymous records.

3.2 Proposal

We take our example at the very beginning of the ownership chain with the sell of a device D. We consider the following actors : the device's manufacturer M, and a retailer R that wishes to acquire D for its store. Following Assumption $A4$, both M and R possess a blockchain address, the corresponding public and private key pair, and the means of submitting or retrieving a transaction to or from the blockchain.

Field	**Description**	Status
Tx type	Possible values are *genesis* and *transfer*	*Mand*
Nounce	Can be made mandatory for *genesis* tx (see Section 3.3)	*Opt*
Inputs	Lists all the inputs of the tx (see Table 4)	*Opt*
Outputs	Lists all the outputs of the tx (see Table 5)	*Mand*

Table 3. Transaction format

Field	**Description**	Status
Previous tx	*Hash* of a previous tx	*Mand*
Index	Index of output to be used in previous tx, must be unspent	*Mand*
Public key	Public key that matches the address that owns the selected output	*Mand*
Signature	Must be signed with the private key that matches the given public key	*Mand*

Table 4. Input format

The general idea is to link the asset's exchange to a series of blockchain transactions, thus creating a chain of ownership. There are two types of transaction available. The transaction that creates the link between the asset and its digital

Field	Description	Status
Destination	Blockchain address of the output owner	*Mand*
Secret	See Section 4	*Opt*
Data	See Section 5	*Opt*

Table 5. Output format

counterpart is the *asset genesis transaction*. Transactions that mark a change in ownership are *transfer* transactions. Transactions follow the Bitcoin [17] model of input/output, meaning that each transaction uses previous transaction outputs as inputs. Transactions are detailed in Table 3. Table 4 and Table 5 breakdown the construction of inputs and outputs respectively.

For an input to be valid, it must be signed with the key corresponding to the output's destination address: transaction tx_0 has 2 outputs, out_0^0 sent to $addr_A$ and out_1^0 sent to $addr_B$. Transaction tx_1 uses out_0^0 as input. To be valid, the input must carry the public key corresponding to $addr_A$ along with a valid signature produced using $priv_A$, private key corresponding to $addr_A$. Because outputs only carry blockchain addresses, and because hashes are irreversible, the public key pub_A is needed for the signature validation (reminder: $addr_A = Hash(pub_A)$. A *genesis*-type transaction has no input. Each output in a transaction corresponds to a different asset.

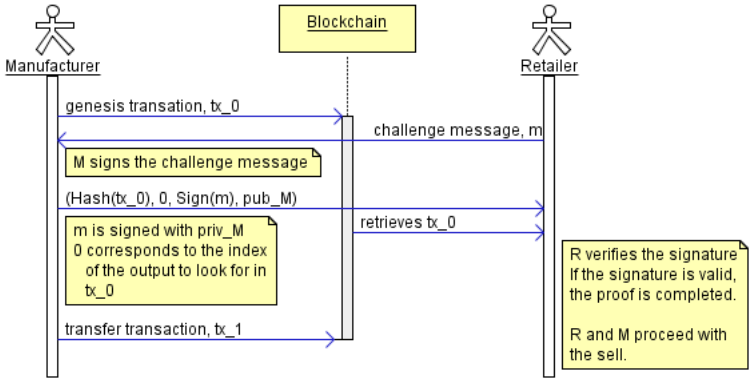

Fig. 1. Proof of ownership and ownership transfer

Going back to our example, M issues a *genesis* transaction, tx_0, with a single output sent to $addr_M$, her own blockchain address, creating device D's digital representation and registering herself as the original owner. Before a sell can take place, M must produce a valid proof of ownership to R. To this effect, R sends a challenge message, m to M. Because the challenge is chosen by R, M cannot reproduce an intercept message. M signs the message with $priv_M$ and sends

$(Hash(tx_0), 0, Sign_{priv_M}(m), pub_M)$ back to R. Using the transaction identifier, R can check for herself the corresponding transaction in the blockchain. She validates that out_0^0 is not spent and was addressed to M. R then verifies the provided signature. If the signature is valid, the proof of ownership is accepted. It is interesting to note that providing such a proof does not compromise the owner's pseudonymity.

Following Assumption $A1$, blockchain keys cannot be stolen. Assumption $A2$ states that the blockchain's cryptographic primitives cannot be broken. This means that the only person capable of producing a valid proof of ownership is the owner of both the blockchain key and the device. Furthermore, since the blockchain history cannot be altered according to Assumption $A3$, once an ownership record has been published or updated, it cannot be modified. This neutralizes Threat $T7$.

Now that M has proven he is the rightful owner of D, the sell can proceed. When R purchases the device, M issues a second transaction of type *transfer*, tx_1. Transaction tx_1 has one input, out_0^0, signed with $priv_M$, and one output, out_0^1 sent to $addr_R$. This second transaction transfers the ownership of D to R.

3.3 Limitations

Blockchain-related limitations The principal limitation of this solution is that the security of the scheme depends on the security of the underlying blockchain [16]. Amongst other issues we can cite 51% attacks, propagation delays [11], withholding attacks [10], the untested scalability of blockchains, their complex governance system, etc. Assumption $A2$ does not cover these issues as they are not crypto-related but rather network-related. However, despite all these theoretical shortcomings, blockchains like Bitcoin and Ethereum have demonstrated their resilience to attacks and only grown stronger as a result.

Another issue that needs addressing is the resistance to DDoS attacks. In the Bitcoin blockchain, the only transactions without inputs are *coinbase* transactions. First transaction of a block, a *coinbase* transaction can only be issued when a block is mined. Furthermore, now that Bitcoin miners' payment is moving from block reward to transaction fees, all other transactions have a cost. This mitigates DDoS attacks as the cost is linear in the number of transactions. When a big number of transactions floods the network, miners can temporarily increase transaction fees, thus rendering an attack even more costly. This last argument still applies to our use case. Because *genesis* transactions are not exempt of fees, trying to flood the network with them has a cost that is at least linear in the number of transaction and can even grow faster as the miners's fees adapt to the situation. This addresses Threat $T5$. Valid transactions can also be created by transferring a device's ownership to oneself. But the cost is the same.

We propose two additional means of mitigation. The first solution is to use a private blockchain where the right to issue *genesis* transactions is limited to pre-approved actors. Manufacturers would need to get registered and only them could then create new devices. A manufacturer that behaves incorrectly, by advertizes non existing devices or issuing too many *genesis* transaction, would lose

it's publication privileges. This has the added advantage of addressing Threat $T8$. Private blockchains unfortunately do not offer the same openness and decentralization as public ones.

A second solution consists in increasing the cost of *genesis* transactions. They would require a *nounce* as an input. Similarly to Proof of Work, the *nounce* would be chosen so that the hash of the transaction is lower than a pre-defined threshold. The difficulty does not need to be as high as Bitcoin proof of work's and can be adapted to counter DDoS attacks. The downside is that this increased computational cost will mostly impact manufacturers. This is therefore likely to impact the device's cost in return.

Use-case-related limitations In the above proposal, a *genesis* transaction creates the digital representation of a device. However, no proof of the existence of this device is required. The production of a valid proof of ownership does not translate to the possession of a real-life IoT device. In case of theft for instance, the owner can still produce a valid proof but will not be able to produce the device itself. This situation is not different from online shopping where the buyer has to rely on pictures, listings, reputation, or other criteria to decide whether to trust the vendor. To strengthen this link between digital and tangible, the *data* field of *genesis* transactions' outputs can be used to specify the device's identifier. This addresses Thread $T8$.

Similarly, the issuance of a *transfer* transaction does not force the shipping of the device to the new owner. It means however that the previous owner can no longer prove that she owns the device. This is a deterrent as future prospective buyers are unlikely to commit to the sell if the ownership cannot be proven. Bitcoin multisignature can be used to solve this problem. This refers to transactions that need more than one signatures to be valid. The desired number here is 2 out of 3. The buyer and vendor choose a party that they trust to be impartial. The buyer then sends the funds to the multisignature address. If everything goes smoothly, upon reception of the purchased item, the buyer and seller both sign the transaction and funds are sent to the vendor. When a conflict occurs, the third party decides who should receive the funds and signs the transaction together with the interested party. The same can be done with ownership transaction. This addresses Threat $T2$.

An owner could also try to sell the same device to two different people. This is a problem that is similar to the cryptocurrency double spending. In a similar fashion, both transactions cannot co-exist. New owners should therefore be sure to wait for the blockchain transaction to be confirmed. For Bitcoin, the generic rule is to wait for the transaction to be burried under 5 to 6 blocks, which takes around an hour. For such a use case, this delay is not an inconvenience.

Finally, malicious previous owners might want to retain control of their former device after it has been shipped to its new owner. To protect against this risk, the device should be wiped clean upon reception and all credentials should be changed. This addresses Threat $T1$. The same applies to a former device owner that want to prevent her sensitive data from being accessed by the new

owner. Before the device can be shipped, it should be restored to factory default. This addresses Threat $T3$. The necessary steps should also be taken to revoke the device's access to all sensitive services such as a smart home private network. This addresses Threat $T4$.

4 Key management

4.1 Motivation

Security rests on the sharing of secrets. Regardless of the scheme one uses to secure IoT applications, some security bootstrap is required. Certificates need to be created and installed, keys need to be distributed, etc. In short, secrets need to be exchanged to secure communications or encrypt data. When a device is manufactured, initial secrets are provisioned to start the security chain. This means that before using a factory-fresh device, the initial secrets need to be retrieved from the manufacturer. The means currently at our disposal to do so lead to slow and cumbersome deployment processes. What is needed therefore is a mean of efficiently retrieving that information in order to be able to remotely and efficiently configure devices in an industrial context.

Currently, physical access to the device is often necessary. When buying a device, the new owner will have it shipped to her location and configure it. The pin or the password may be written down on the device's box, in the configuration manual or otherwise physically attached to the device and its packaging. Buyer and seller might also choose to call on a trusted third party to take care of the configuration and installation of devices.

The need for an initial physical access is a hindrance on the deployment process. Because many devices need to be configured at once, this method that is slow, costly and may require to trust confidential information to a third party is ill-fitted.

Another issue is the management of these secrets. In IoT scenarios, multiple devices may be owned by the same entity. Furthermore, symmetric cryptography is often preferred due to the constrained nature of IoT devices. This is another multiplying factor for the number of keys involved. This multiplicity implies the need for an efficient management of secrets over the life a device. Based on the blockchain ownership records, we propose a solution that both delivers a device's secret to its newest owner and allows her to manage such secrets over the life of the device.

Benefits of the proposal The benefits brought by the proposed scheme are as follows:

- *Simplified deployment process*
- *Cost reduction*
- *Reduction of the number of secret keys*
- *Distributed storage of keys*

As an extension of the proposal from Section 3, to the benefits described above we add the advantages described in Section 3.1 and are inherent to the use of a blockchain as the underlying mechanism.

4.2 Proposal

Once again, we start at the beginning of the ownership chain. The manufacturer M sells a batch of n devices $\{D_i\}_{0 \leq i < n}$ to a retailer R. Each device has a unique identifier id_i. Additionally, R owns a master key K_R. The symmetric key $k_{R,i}$ is derived from K_R and id_i. Key $k_{R,i}$ will be used to encrypt s_i, secret linked to device D_i. The blockchain still supports two types of transaction, *genesis* and *transfer*.

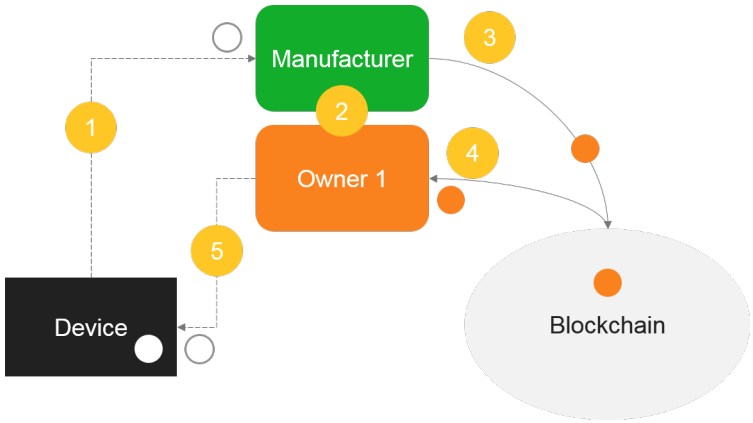

Fig. 2. Transferring ownership and delivering device secret

Figure 2 illustrates the process by which ownership is transferred and secrets are exchanged :

Step 1 M interacts with each D_i and retrieves a secret s_i. This secret s_i can be an administrative password, a private key, a pin, etc.

Step 2 M retrieves $\{k_{R,i}\}$ from R. That information can be provided along with payment information for instance. Note that when all devices are purchased from the same seller, it is not necessary to use different $k_{R,i}$. If R prefers using asymmetric cryptography, the key used to encrypt s_i can also be retrieved from a registry storing public key records. It is used for applicative purposes and should differ from the keys used for the blockchain protocol. Using $\{k_{R,i}\}$, M encrypts each s_i.

Step 3 M issues a *genesis* transaction, tx_0, with n outputs where the out_i^0 corresponds to D_i and is sent to $addr_M$, her own blockchain address. M issues a second transaction of type *transfer*, tx_1, with $\{out_i^0\}$ as inputs, signed with

$priv_M$. This transaction yields n outputs, one for each D_i, sent to $addr_R$. In addition to $addr_R$, each output carries $Enc_{k_{R,i}}(s_i)$.

Step 4 R retrieves $\{Enc_{k_i}(s_i)\}$ from the blockchain and deciphers them, recovering $\{s_i\}$.

Step 5 Using the s_i, R gains access to and configures if need be each D_i.

The same process can then be repeated by the new owner to sell the device to somebody else. tx_1's outputs can be separated allowing for devices to be sold separately.

This scheme involves a lot of keys and secrets but only K_R and $priv_R$ need to be safeguarded by R. Each s_i can be recovered from K_R. This greatly simplifies the management of secrets where many devices are involved.

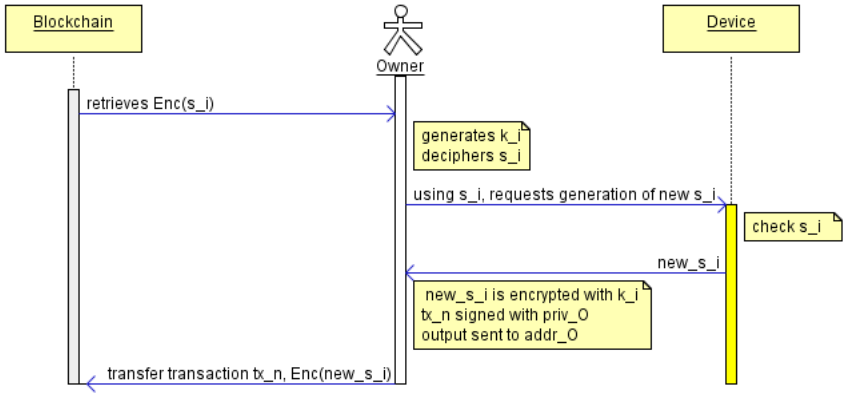

Fig. 3. Publishing a new secret to the blockchain

Furthermore, updates can be made to a device's secret, as illustrated by Figure 3. After buying a device D_i, the new owner should change the corresponding s_i as this secret is known to the previous owner (Threat $T1$). Using s_i, she interacts with D_i and generates a new secret, s_i^{new}. This secret can now be stored in the blockchain. The owner can simply further the ownership chain by sending a *transfer* transaction to herself, replacing $Enc_{k_{R,i}}(s_i)$ by $Enc_{k_{R,i}}(s_i^{new})$, where Enc is the encryption algorithm. Such a transaction can also be made to hide the link between the owner's identity and her blockchain address (Threat $T6$).

4.3 Limitations

The first delicate point of this scheme is the transmission of the encryption key(s), $\{k_{R,i}\}$, from R to M. If symmetric keys are used, then a secure communication channel should be put in place to allow for the exchange. When a large number of sells involve the same actors, typically M and R, the same key can

be used for all encryptions. That way, the symmetric key must be exchanged only once. Another work around is to use asymmetric cryptography. The public key can then be transferred without the need for particular precautions. This mitigates Threat $T6$.

When the $\{k_{R,i}\}$ are symmetric keys, they should of course be changed before a new secret can be uploaded to the blockchain. Otherwise, all s_i^{new} are exposed to M (Threat $T1$). To achieve this, $k_{R,i}$ can be derived from elements linked to blockchain transactions. Let tx_n be the latest transaction that proves M owns D_i. Such a transaction must exist with unspent output otherwise M is not D_i's rightful owner. In that case, $Hash(tx_n)$ could be used as an input for key derivation. The *transfer* transaction from M to R, transaction tx_m, will use one of tx_n output as input. Similarly, when updating s_i, an output from tx_m will be used. The hash of the previous transaction is therefore an easy element to recover. It varies with transactions, leading to different $k_{R,i}$.

5 Sharing configurations

5.1 Motivation

We have seen that information pertaining to the device can be stored into the blockchain for its owner's use. It is also the case for information that would be used by users or other IoT devices willing to interact with it. The first example that comes to mind is the advertizing of a public key. Contrary to static properties such as memory space or power consumption, a device possesses a number of dynamic properties, its public key being one of them. Such properties can be useful to others for any number of reasons such as authentication, evaluation of trust (and risks), discovery of assets, etc. Other examples include advertizing known protocols, the version of the softwares that are running on a device or other application-specific information.

During the life of a device, the owner can share and update that information using the blockchain. Transfer transactions can be assorted with any number of properties that the owner judges relevant to the use of her device. Updates are made by transferring ownership to oneself. The users can then retrieve that information from the blockchain.

Benefits of the proposal The proposed scheme presents the following advantages:

- *Distributed storage*
- *Shared Infrastructure*
- *Availability*

5.2 Proposal

Let O be the owner of a device. O wishes to publish $Prop_D$, set of dynamic properties of D. Let U be a potential user of D. User U needs to consult $Prop_D$ before exchanging with D. Let tx_n be the latest blockchain transaction designating O

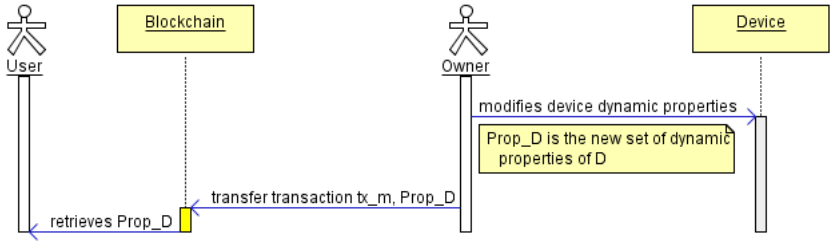

Fig. 4. Publishing dynamic properties to the blockchain

as D's owner. The corresponding output, out_l^n, must be unspent, otherwise, O is no longer the owner of D.

As illustrated in Figure 4, O issues a *transfer* transaction, tx_m, with out_l^n as input, signed with $priv_O$. This transaction yields 1 output sent to $addr_O$ that, additionally, carries $Prop_D$. U can then retrieve tx_m and extract $Prop_D$.

5.3 Limitations

The first question that comes to mind is the alloted size for $Prop_D$. This, once again, depends of the underlying blockchain. In Bitcoin, the maximum size of a block is fixed at 1 MB[7]. Naturally, this limits the size of a transaction. Furthermore, bigger transactions have a higher cost. Since miners are paid by transaction fees, if transaction fees are fixed, they would rather include many small transactions as that will amount to more fees. To compensate for that, bigger transactions should pay a higher fee. In Ethereum, there is no fixed limit to the size of a transaction but the amount of gas per block is limited. Even if this limit augments with time, the size of a transaction is currently limited to around 100.000 non-zero byte. Gas being paid for with ether, bigger $Prop_D$ also amount to higher cost. Published data should therefore be kept to a minimum. This is inline with many other IoT requirement however.

Privacy concerns might arise from the publication of device information in such public fashion (Threat $T6$). They can be tackled by either using a private blockchain or encrypting the published content to restrict its viewing. Group keys [13] or attribute-based encryption [12] can be used to efficiently control access to that information.

Finally, IoT devices may want to retrieve $Prop_D$. Unfortunately, the resources required to maintain a connection to the blockchain network are too much for constrained devices at the moment. This limitation is not due to the blockchain technology but rather to its youth. Light clients specifically designed for IoT devices should emerge before long.

[7] Many want to increase this limit but this would require a hard fork. The issue is still being debated.

6 Conclusion

The blockchain has made the tracking of asset's ownership relatively inexpensive. It does not have to be reserved for houses and boats anymore. We therefore propose to use it to track the ownership of IoT devices. The chain of ownership can be augmented by adding additional information to transfer transactions. We present two ways to do so. Firstly, information can be added to help the owner manage its devices. Secondly, information can be added to inform users and other devices of a device's dynamic characteristics that can be relevant in their exchange.

We have defined four security assumptions and eight security threats involving owners and third parties. Limitations of our proposals have been argued. Currently, the requirement that every potential owner possesses a blockchain address is the most unlikely. This is likely to evolve in a near future. The fact that blockchain keys cannot be lost or compromised is also a big assumption. However, the issue of safekeeping a key has been studied extensively and many solutions can be provided.

References

1. Arcade City. `https://arcade.city/`. Last checked Feb, 16th 2018.
2. Exodus. `https://www.exodus.io/`. Last checked: February, 23th 2018.
3. First blood. `https://firstblood.io/`. Last checked Feb, 16th 2018.
4. Follow my vote. `https://followmyvote.com/`. Last checked Feb, 16th 2018.
5. Melonport. `https://melonport.com/`. Last checked: February, 23th 2018.
6. Nxt. `https://nxtplatform.org/`. Last checked: February, 23th 2018.
7. Vitalik Buterin et al. Ethereum white paper. `https://github.com/ethereum/wiki/wiki/White-Paper`, 2013. Last checked : 23/09/2016.
8. Konstantinos Christidis and Michael Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.
9. Marco Conoscenti, Antonio Vetro, and Juan Carlos De Martin. Blockchain for the internet of things: A systematic literature review. In *Computer Systems and Applications (AICCSA), 2016 IEEE/ACS 13th International Conference of*, pages 1–6. IEEE, 2016.
10. Nicolas T Courtois and Lear Bahack. On subversive miner strategies and block withholding attack in bitcoin digital currency. *arXiv preprint arXiv:1402.1718*, 2014.
11. C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE International Conference on Peer-to-Peer Computing*, pages 1–10, 2013.
12. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98. Acm, 2006.
13. Hugh Harney and C. Muckenhirm. Group key management protocol (gkmp) architecture. RFC 2094, RFC Editor, July 1997.
14. Seyoung Huh, Sangrae Cho, and Soohyung Kim. Managing iot devices using blockchain platform. In *Advanced Communication Technology (ICACT), 2017 19th International Conference on*, pages 464–467. IEEE, 2017.

15. Nir Kshetri. Can blockchain strengthen the internet of things? *IT Professional*, 19(4):68–72, 2017.
16. Iuon-Chang Lin and Tzu-Chun Liao. A survey of blockchain security issues and challenges. *IJ Network Security*, 19(5):653–659, 2017.
17. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
18. Kyosuke Osaka, Tsuyoshi Takagi, Kenichi Yamazaki, and Osamu Takahashi. An efficient and secure rfid security method with ownership transfer. In *RFID security*, pages 147–176. Springer, 2008.
19. R. Pass, L. Seeman, and A. Shelat. Analysis of the blockchain protocol in asynchronous networks. 2016.
20. Biplob R Ray, Jemal Abawajy, Morshed Chowdhury, and Abdulhameed Alelaiwi. Universal and secure object ownership transfer protocol for the internet of things. *Future Generation Computer Systems*, 78:838–849, 2018.
21. Evangelos Rekleitis, Panagiotis Rizomiliotis, and Stefanos Gritzalis. How to protect security and privacy in the iot: a policy-based rfid tag management protocol. *Security and Communication Networks*, 7(12):2669–2683, 2014.
22. Meni Rosenfeld. Overview of colored coins. *White paper, bitcoil. co. il*, page 41, 2012.
23. Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.
24. Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 2014.