A Model for Signatories in Cyber-Physical Systems

Sreelakshmi Vattaparambil Sudarsan
Luleå University of Technology
Luleå, Sweden
sreelakshmi.vattaparambil.sudarsan@ltu.se

Olov Schelén
Luleå University of Technology
Luleå, Sweden
olov.schelen@ltu.se

Ulf Bodin
Luleå University of Technology
Luleå, Sweden
ulf.bodin@ltu.se

***Abstract*—Distributed Internet of Things and cyber-physical systems can potentially be used as agents to automatically sign events and transactions on behalf of users. To accomplish this, there is a need for a model that can represent the relationships, credentials and organizational hierarchies of people and devices, facilitating agents acting as signatories in a controlled way. This paper proposes such a model, where people in different positions are entitled to sign on behalf of organizations or departments therein and extend that to representing machines. Central in this model is the Power of Attorney (PoA), which is a self-contained and signed digital document that for a limited time and in a defined context, authorizes a particular agent (whether a person or device) to sign on behalf of a principal. Although such self-contained PoAs can be stored anywhere, we propose a conceptual architecture based on PoAs and include a signatory registry that keeps track of organizational hierarchies in terms of people and devices according to the defined model and stored PoAs in that context.**

***Index Terms*—Signatory, Power of Attorney (PoA), Cyber-Physical System (CPS), Certifying Authority (CA), Public Key Infrastructure (PKI), Internet of Things (IoT)**

I. Introduction

The digitization and automation of interactions between parties representing legal entities often require the actions to be authorized and signed before they can be executed. Assume, for example, that you want someone to be your proxy for picking up your package at the post office or prescription medication at the pharmacy. A common requirement (depending on the country) for them to deliver the items to a proxy is that you sign the delivery notice and provide your identification and a Power of Attorney (PoA) that entitles your proxy to pick up this specific item. The proxy must be able to reference the signed delivery notice, your ID, the PoA, and the ID card of himself/herself at the delivery point. Today, physical papers are entirely or partly used so that the proxy brings your ID card, the signed delivery note and PoA as well as his/her own ID card to the delivery point.

In a machine scenario, this has to be completely digitized and automated for seamless operation. We address devices that are powerful enough to have a digital identity and carry out public and private key authorization autonomously; i.e., in this paper, we do not address resource constraints in devices. Assume, for example, that you want your autonomous car to be your proxy. Both you and your car need strong identities, and you need to assign a PoA to your car, which is valid for the specific purpose and often for some limited time. When such generic mechanisms are in place, they can be used in many situations, for example, to provide generic powers to engage on your behalf in specified actions, within certain limits (e.g., of value) and within a limited time. Such a mechanism can also cover simple payment schemes where a device obtains some money that can be used for predefined purposes. An example is that your autonomous car obtains some prepaid amount of currency to use for passing through tolls (micropayments).

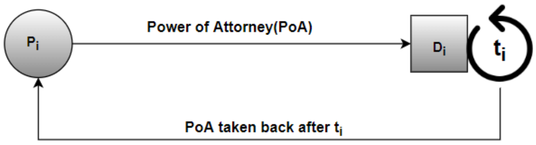

Fig. 1. **Power of Attorney**

In such a machine scenario, it is important to have full control of the devices, both in terms of ownership and of which powers they are given and by whom, especially if they are to assume powers that are not provided by the owner.

In this paper, we introduce a conceptual model of signatories, where devices sign for people using the PoA of the user for a user-defined period; see Fig. 1. The PoA is a signed document showing user ownership and directing it to an authorized second party. Digital signatures (DSs) are used in the signing process. The DSs are implemented by public key cryptography because of some key properties: [1]

- The receiver can authenticate the sender of the message.
- Provide nonrepudiation.

An example is to give your device a PoA to buy something at a given shop, within a limited time, for a maximum amount of money and to be invoiced later.

In an organizational hierarchy, people at specific hierarchical levels are allowed to sign for a particular department or division in the company. This hierarchical structure plays an important role in every organization due to its flexible management. The decision making and approval of various documents are done by assigned roles in the organization. As far as the organization is concerned, document signing is a very important task that requires high-level security and management strategies. This paper proposes a mechanism where devices can sign on behalf of humans or individuals, thus bringing the management structure to a new level. This secure device signatory approach provides an automated solution for

978-1-7281-8956-7/20/$31.00 ©2020 IEEE

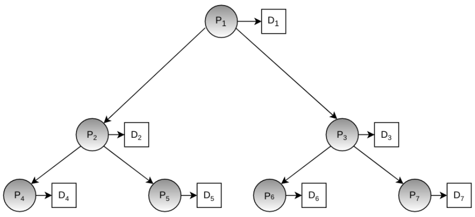

Fig. 2. **Hierarchical structure**

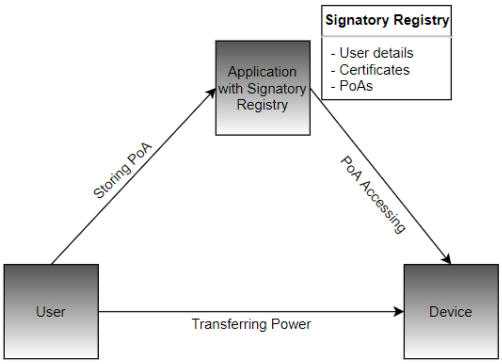

Fig. 3. **PoA Storage and Access**

the entire signing process. Here, it is subdivided into several hierarchical levels (Fig. 2).

II. BACKGROUND CONCEPTS

A. *Power of Attorney (PoA)*

PoA is the official document that the signatory signs and directs to the receiver so that the recipient can use the power to execute user actions. Here, the owner is called the principal, and the device that receives PoA is called the agent or attorney. PoA is the term for powers of representation [2]. Granting power does not always require a written document. However, certain countries use a written document as proof. This legal document should define all the restrictions on power transfer, and the agent should have actual knowledge of those restrictions. PoA is not privative, and the principal has complete authority over it. It is also in practice to grant PoA to multiple parties so that every PoA is allowed to function independently in this scenario. A PoA may or may not be transferable by an agent to another second party, who is considered a substitute attorney. Both the principal and the agent use certain authorization processes in the existing legal framework to verify one another's identities. In this work, we use a signatory registry for individual authentication and data storage, as shown in Fig. 3.

B. *Certifying Authority (CA)*

The certifying authority plays a significant role in the security authentication process [3]. Authentication is the process of proving the authenticity of the sender to prevent the

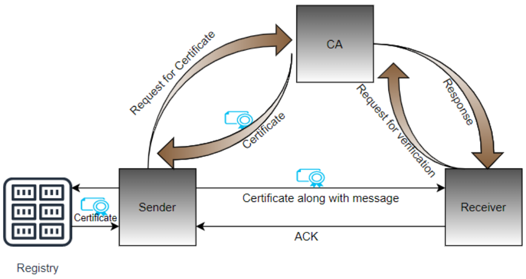

Fig. 4. **Certifying Authority**

repudiation of message transmission. The public key, which is openly accessible to everyone in the network, is used to uniquely identify everyone in the network. The public key is usually used to identify the sender of the message in message communication. However, many malicious attacks based on public keys, such as public-key modification and the replacement of genuine public keys with those of attackers, have been reported. All of these factors result in the need for public-key security, achieved through public key infrastructure (PKI). The CA is a trusted third party who can issue public key certificates that contain the user's public key. Initially, the user sends his/her credential information to the CA along with his/her public key, as shown in Fig. 4. The CA verifies all the submitted documents and issues the public key certificate for the user for a while. Public keys are certified by a signature based on the private key of the CA using the evidence collected from the owner. In this work, we use the public key certificate to protect the user's public key. We create this exclusive public key certificate by sending the user ID and other credentials along with the public key to the CA, which is then stored in the signatory registry along with other user credentials. Hence, during each signing process, the signatory uses his/her public key certificate to represent his/her identity. The receiver uses this same certificate to cross-check with the CA and confirm the sender's authenticity. The active revocation of certificates is not included in our work.

III. Conceptual Model

The model defines two different structures: PoA, which is completely generic and self-contained, and the hierarchical organization, resembling real-world situations.

The organization is defined by a hierarchical structure (i.e., a directed acyclic graph) of actors (i.e., people and devices) in our proposed scheme. Here, the head of the organization is indicated with the root node in the hierarchy. All other nodes represent other department and division heads, and other normal employees and devices are indicated using the leaf nodes in the tree structure. Fig. 2 represents the organizational hierarchy with device association.

The hierarchical organization can be enforced by PoAs. Users at the higher level can issue PoAs to both users and devices at the lower levels and complete the work via devices.

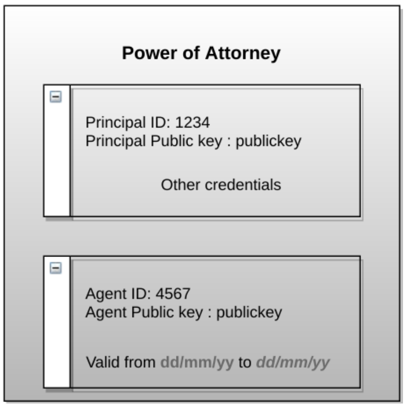
Power of Attorney
Principal ID: 1234
Principal Public key : publickey
Other credentials
Agent ID: 4567
Agent Public key : publickey
Valid from dd/mm/yy to dd/mm/yy

Fig. 5. **PoA Document Content**

Additionally, with a specific PoA, anyone can potentially authorize anyone else to sign on his/her behalf.

The model is highly concerned about the security issues of accessing data. The signatory registry stores all the information regarding the model for a particular organization. Each actor, authorized by his/her private key, uses data stored in the signatory registry. Hence, the principal may use the signatory registry to verify the agent's identity when issuing a PoA. The signatory registry, which is an extended CA, hence plays an important role in this work, although self-contained PoAs can exist and be used without a signatory registry. When used, the registry keeps track of public key certificates, PoAs and organizational structures.

a) PoA Usage: PoA contains the public key of both principal and agent, expiry date and other credentials; see Fig. 5. In our work, the PoA is signed using the private key of the principal for a specific time. It is assigned to the delegated agent (person or device). Thus, every device with a PoA becomes associated with a person, and the device public key is generated by the owner, who signs it using his/her private key.

PoAs are not secret documents. The agent has to authenticate himself/herself upon receiving the PoA. After a specific time, the PoA will become invalid and can no longer be used by the receiver agent. A benefit of this is that stale PoAs will not remain active. A PoA can be reissued in a soft state approach when an eternal PoA is desired. All the PoAs are also stored securely in the signatory registry along with other user credentials, similar to public key certificates.

The organizational hierarchy provides a generic model of signatories (Fig. 2). In addition to businesses, it can also be used by private people and their devices, although the hierarchy in that case typically has very few levels. Here, Person P_i is associated with device D_i. The person is uniquely identified with a pair of keys: public and private keys. The same is applicable for devices, which makes them unique in the device network. The person can use the device to sign on behalf according to his/her needs. The signed PoA is sent to device D_i from person P_i and remains active for period t_i. This time period is decided by the person. After a certain calculated time period, the device will automatically lose the power of authority. The following assumptions are made:

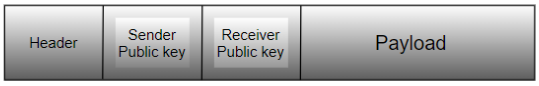

Fig. 6. **Message Format**

- Public and private keys are made by the user using secure cryptographic algorithms.
- Each person or device has one or many self-generated private-public key pairs for different contexts, and the private key is stored securely by the person or device.
- We do not address the threat of identity theft, e.g., by stolen private keys. Preventing identity theft is a separate research topic.

b) Document Signing: In this paper, we use the term document to represent a contract, a transaction, or any other kind of event that needs to be signed. Document m_j to be signed has a flag bit that indicates the public key PK_{pj} of the person who is intended to sign the document (signatory) and the sender of the document. By referring to this field in the message in Fig. 6, the signatory can identify the message that has to be signed. According to the organizational hierarchy system, the higher authority can sign for all divisions that come under their department. Hence, if document m_j is intended for position P_i, then anyone in a position higher up in the hierarchy can sign m_j, which means that it can be signed by all P_j, where j $\leq$ i. This is applicable in the scenario of multiple signing requirements, where the document requires signatures from multiple people. In this case, the document is directed to the intended recipient by decrementing the recipient flag bit in the message format.

Similarly, the message can be signed using a device associated with the person according to his/her unavailability. In this case, the PoA is issued for device d_j for time period t_i. Similar to the person hierarchy, m_j can be signed by all other devices that are above d_j in the hierarchy, which means that m_j can be signed by all d_j, where j $\leq$ i. Algorithm 1 describes the device signing process.

We assume that the owner generates a pair of public and private keys using a secure cryptographic algorithm. After the creation of his/her key pair, the user will generate a public and private key pair for their device. This will be signed using the users' private key to approve the pair of device keys. Finally, both the device and user credentials, including the public key, certificates and PoA, are stored securely in the signatory registry. Now, the device can actively participate in the signing process. The flowchart is shown in Fig. 7. Every time, the device checks the availability of PoAs; if a PoA is available, then it will go for the signing process after checking the validity of the PoA. If the PoA is expired (invalid), then the device can request the owner for a new PoA.

c) Signatory Registry: Authentication plays an important role in the signatory hierarchy system. Here, each person and device are authenticated using their public key. The public key

Algorithm 1 Device Signing

```
Require: $M = m_1, m_2, m_3...m_j$ set of documents, $P = p_1, p_2, p_3...p_j$ number of people, $D = d_1, d_2, d_3...d_j$ set of devices, $T = t_1, t_2, t_3...t_j$ time period, $PK$: public key, $K_{prvt}$: private key, $\alpha$: group of messages to be signed, $\phi$ : $P_i$ is available for signing, $f(P_i,\ m_i)$: signing function, where $P_i \in P$ and $m_i \in M$.
Ensure: $m_i$ signed document.
if $m_i \in \alpha$ AND $P_i \in \phi$ then
    $f(p_i,\ m_i\ )$
else
    if $P_i \notin \phi$ then
        $p_i$ calculate $t_k$
        $d_i \Leftarrow p_i$ for $t_k$
        $f(d_i,\ m_i\ )$
    end if
    return $m_i$
end if
```

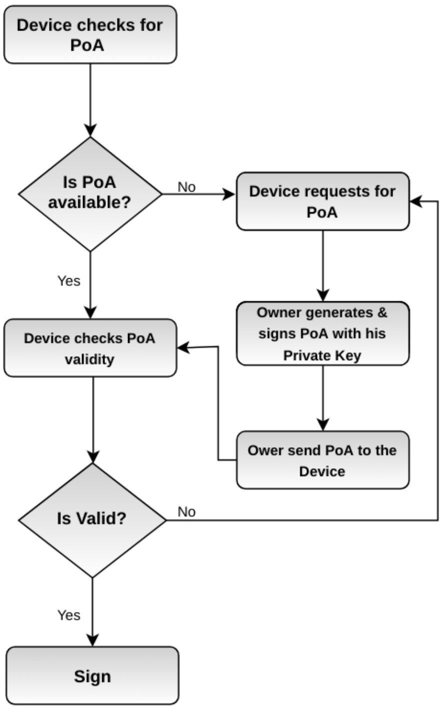
Device checks for PoA
Is PoA available?
No
Device requests for PoA
Yes
Owner generates & signs PoA with his Private Key
Device checks PoA validity
Ower send PoA to the Device
Is Valid?
No
Yes
Sign

Fig. 7. **Signaling Diagram for the Example**

and other useful user data are stored in the signatory registry. The signatory registry is primarily implemented to add and look up the database for the various users registered in the system, and the signatories for each level are identified with their public key, which provides a hierarchical level of signatories and their personal information. It is always convenient to store everything in an organized manner. It is implemented as a database with various fields to store the personal information of each user or device registered to the system. Based on various parameters, such as the CAP (consistency, availability and partition tolerance) theorem, level of complexity, and number of relationships, the right database can be chosen. Our preference is to use a relational database to obtain a strong check of foreign key constraints and transactional semantics for future requirements. The database can be modeled based on organizational hierarchy, with users at different levels. There are separate tables for devices, which are designed based on the device hierarchy. The device and user hierarchies are intertwined to ensure that users and devices are synchronized. The model defines two hierarchical sql models:

- Organizational hierarchy
- Device hierarchy

In the organizational hierarchy, the person with a specific signatory ID in an assigned role can sign for the corresponding department and all the departments below it in the hierarchy. The individual departments are identified uniquely with the department ID field in the department table. Here, each person is connected with the fingerprint table with a one-to-many relationship. This means that the person with his/her corresponding ID/public key can be connected with the fingerprint details of the same person. The fingerprint table is used to store the unique details of the people and stores the hash of each person, and the appropriate hash functions are stored in the table for information. The same goes for device hierarchy, which enables machine-to-machine communication.

Upon receiving a message, the person or device checks for the recipient data in the signatory registry using the public key attached to the message. This will ensure that the message or document is sent by an authorized user in the organization. The other scenarios where authentication is needed are as follows:

- Issuing the PoA to the device.
- Sending the document to get signed from one level to the next in the hierarchy.
- Sending the document that requires multiple signatures.

In these cases, the person looks up the signatory registry using the public key of the intended user for authentication purposes. Note that registering a PoA in a signatory registry is optional (comparable to registering a public key in a CA). Hence, the signatory registry may not be used if the parties already trust the PoA and each other's identity.

IV. Related Work

There have been many interesting works done in the field of machine-to-machine communication security, as summarized in Table I. Our contribution focuses on signatories in various systems, such as the cyber-physical system (CPS), where we assign powers to people and devices through PoAs. We also propose a signatory registry to keep track of hierarchical organizations of people and devices. The digital PoAs in combination with the signatory registry and methods for authorization and signing of events are novel. Most related works focus on the authentication and authorization needed as basic building blocks. Therefore, this section covers

TABLE I
COMPARISON OF EXISTING WORKS

Title	Author	Methodology	Findings
A study on application of digital signature technology (2010)	Junling Zhang	Digital signature	Application and implementation techniques of digital signatures in network security
A Lightweight Authentication Mechanism for M2M Communications in Industrial IoT Environment (2019)	A. Esfahani et al.	Crypto hashing and XOR operations	Simple method for the authentication of resource constraint devices
SAKES: Secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6L0WPAN) (2013)	H. R. Hussen et al.	IPSec and Diffie Hellman key exchange algorithm	Authentication mechanisms for IP-based sensors over the 6LOWPAN
GSLHA: Group-based Secure Lightweight Handover Authentication Protocol for M2M Communication (2019)	M. M. Modiri et al.	Group-based handover protocols	New security mechanism using group-based handover protocols using secret key and group ID mechanisms for authentication
A distributed authentication and key exchange approach for secure M2M communications (2017)	B. S. Murthy and L. Sumalatha.	Symmetric encryption	A simple security design for the authentication of sensor nodes using a lightweight public key and symmetric cryptography
Distributed Ledger Technologies for M2M Communications (2019)	N. Zivic et al.	Distributed ledger technology	IoT security using DLT technology and analyzes various BC platforms for IoT compatibility
A Distributed Approach for Secure M2M Communications (2012)	Y. Ben Saied et al.	Public key cryptography	A key establishment solution for heterogeneous machine-to-machine (M2M) communications is proposed
Can Blockchain Strengthen the Internet of Things? (2017)	N. Kshetri	Blockchain	Blockchain for identity and access management system in IoT
Blockchains and Smart Contracts for the Internet of Things (2016)	K. Christidis and M. Devetsikiotis	Blockchains and smart contracts	Discuss various complexities of using BC with IoT and its possible solutions
Blockchain for IoT security and privacy: The case study of a smart home (2017)	A. Dorri et al.	Blockchain	A smart home setting and consists of three main tiers, namely, cloud storage, overlay, and smart home. It shows how the BC-based smart home framework is secure by thoroughly analyzing its security with respect to the fundamental security goals of confidentiality, integrity, and availability
Managing IoT devices using blockchain platform (2017)	S. Huh et al.	Blockchain	Manage IoT devices using Ethereum, a blockchain computing platform. Manage the required keys using RSA public key cryptosystems, where public keys are stored in Ethereum, and private keys are saved on individual devices
Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT(2018)	O. Novo	Blockchain	New architecture for roles and access control in IoT using blockchain to manage different IoT devices globally

alternative approaches and base technologies that have been explored in the field.

The machine-to-machine approach includes different types of components, such as sensors and high-capacity industrial systems, and they are self-organizing networks that are distributed over heterogeneous locations. B. S Murthy and L. Sumalatha [4] propose a simple authentication and key exchange mechanism for M2M communication. The method is simplified with the use of a lightweight public key and symmetric encryption scheme. The major limitations in M2M communication are as follows:

- The communication medium is a radio signal, which can be easily hijacked.
- The sensor nodes have limited storage capacity and computing skills.

Here, the public keys for authentication purposes are issued by a trusted server, which in turn is used by machine-type communication devices (MTCDs). The trusted authority keeps all the public keys by signing it with their private key.

Machine-to-machine communication is one of the basic components of the IoT. N. Zivic et al. [5] combine DLT with IoT, which can be considered the Internet of Trusted Things. The paper discusses three different DLT technologies according to their architecture: blockchain (BC), tangle and hashgraph. These three platforms are compared in various contexts to find a proper architecture for a specific M2M application. For M2M communication, the most important feature required is a low-cost transaction or microtransaction. When we consider the Bitcoin blockchain, the transaction cost is high. The next popular platform is Ethereum, which makes use of smart contracts suitable for M2M communication. However, it is not scalable, and the transaction cost is particularly costly. The IOTA platform applies mainly for IoT applications because toll-free transactions are suitable for IoT. The other platform is hashgraph. This DLT technology is based on the gossip protocol and does not use the proof of work protocol, and the transaction fee is expected to be the lowest. However, hashgraph is only good for a closed, private network.

Decentralization brings more security to M2M communications. At the same time, this leads to the complexity of committing transactions or mining processes by small resource-constrained devices. A distributed novel approach in which the less resource constraint node obtains assistance from its more powerful neighbors in the network is reported. They discuss situations where a high resource constraint sensor device shares data with a server with a large amount of computing power. This situation is an example of communication between heterogeneous components in a network. The key handshake authentication mechanism using the session key is used in this approach. Symmetric key cryptography was used earlier for secure key establishment and communication in sensor nodes because of the poor computing power of the sensor nodes. Later, with the implementation of lightweight public key cryptography algorithms, sensor networks started using public key cryptography [6].

A study on BC for identity and access management systems in the Internet of Things is carried out to uniquely identify IoT devices. Here, a separate private BC is created and stores cryptographic hashes of individual device firmware. This system protects the sensor network from IP spoofing and IP address forgery attacks. They print out problems with cloud storage, such that if an IoT device connected to the server is breached, all other devices connected to the server could be infected [7].

The BC-IoT combination has been discussed by K. Christidis and M. Devetsikiotis [8], who conclude that BC is a powerful security mechanism for the IoT and can cause significant transformations across several industries. The paper broadly defines how BC and smart contracts work and how they can be integrated with the IoT. According to them, the cloud has a high maintenance cost for the IoT network. The paper discusses various real IoT systems using BC for security, such as Slock.it, an electronic door lock, TransActive Grid, and a renewable energy grid in Brooklyn, NY. The above authors also discuss various complexities of using BC for IoT and its possible solutions. The proof of work protocol of BC can cause scalability issues and can be solved using a sharding network. Second, maintaining privacy in BC is a complicated issue due to its transparent nature in regard to the IoT. The other complexities mentioned are deciding on the minor set, limited legal enforcement of smart contracts and the negative effects of complete autonomy.

The secure BC-based smart home framework is explained, where the smart home has a supercomputing device called a "miner" that handles all communication within and external to the home. This miner preserves a private BC, which provides secure access control for IoT devices in the smart home. The core components of the smart home are transactions, private and local BC, home miners and local storage. Here, devices communicate with each other after sharing a secret key among themselves, which is issued by the miner. The paper analyzes that the overheads incurred by their method are low and manageable for low-resource IoT devices [9].

The technique mentioned by S. Huh et al. [10] is about managing IoT devices using the BC platform. They use the RSA public-key cryptosystem to manage keys and store all public keys in the Ethereum BC by writing smart contracts and private keys on individual devices.

According to O. Novo [11], a new architecture for roles and access control in the IoT using BC to manage different IoT devices globally is proposed. Here, the access control information is processed and distributed using BC. Here, a node called a management hub collects all the information from the IoT devices, and neither the management hub nor IoT devices are part of the BC. It is not practically feasible to run BC on every single IoT device.

The related work presented herein covers authentication and authorization with devices of different kinds and capacities but does not address concepts similar to the PoA and signatory registry. Similarly, the work related to blockchains and ledger technologies does not cover the rights to act on behalf of others.

V. Discussion and future work

Our findings suggest that machines can be used to sign reliably on behalf of a person with the use of PoAs. This automated signing solution using digital PoAs effectively automates the machine economy. Cryptographic authentication and signing can be achieved by a standard public/private key approach as appropriate for secure machine-to-machine communication. Additionally, PoA security is achieved through a CA and signatory registry. In our model, PoAs are the main emphasis, which makes the system do things more effectively with the person's actual power. To avoid misuse, PoAs are strictly regulated by the owner who generates them for the device.

We have presented a generic concept of PoAs and signatory registry that are widely applicable. Although the presented model for digital PoAs, the signatory registry and methods for authorization on its own can facilitate the automation of interactions between devices owned by different parties, further work is needed on the integration with established methods and mechanisms for security and trust. For example, either the integration of existing concepts and proposals for trusted servers confirming public keys or the integration with blockchain-like technologies to enable the signatory registry concept to be more distributed and decentralized. The proof of concept will be presented in future work. Additionally, the use of PoAs by resource-constrained devices is for future work.

VI. Conclusions

In this paper, we address a secure model for signatories in a CPS, which is based on device signing on the person's behalf. This concept provides signing power to people and devices through digital PoAs that can be used to effectively automate the machine economy. It also registers the PoAs in a signatory registry that extends the powers of a traditional certificate authority. The signatory registry thereby supports cryptographic authentication and signing based on standard public/private key approaches. This is different from current automated machine-to-machine communication systems, where the machines use their wallet to do things on the user's behalf.

The signatory registry here is used to systematically record both user and device data using the signatory hierarchy method. Using the signatory registry and existing CA system, the security goals, particularly authentication and authorization, are achieved. We believe that this PoA solution will cause a substantial change in CPS for modern society.

REFERENCES

[1] Junling Zhang, "A study on application of digital signature technology," in *2010 International Conference on Networking and Digital Society*, vol. 1, 2010, pp. 498–501.

[2] B. Schuijling, *Representation, Power of Attorney and Mandate*, 03 2016, pp. 117–122.

[3] B. Bellur, "Certificate assignment strategies for a pki-based security architecture in a vehicular network," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, 2008, pp. 1–6.

[4] B. S. Murthy and L. Sumalatha, "A distributed authentication and key exchange approach for secure m2m communications," in *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, 2017, pp. 277–280.

[5] N. Zivic, C. Ruland, and J. Sassmannshausen, "Distributed ledger technologies for m2m communications," in *2019 International Conference on Information Networking (ICOIN)*, 2019, pp. 301–306.

[6] Y. Ben Saied, A. Olivereau, and M. Laurent, "A distributed approach for secure m2m communications," in *2012 5th International Conference on New Technologies, Mobility and Security (NTMS)*, 2012, pp. 1–7.

[7] N. Kshetri, "Can blockchain strengthen the internet of things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[9] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 618–623.

[10] S. Huh, S. Cho, and S. Kim, "Managing iot devices using blockchain platform," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 464–467.

[11] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.