Trustworthy Data-Driven Networked Production for Customer-Centric Plants

Davy Preuveneers, Wouter Joosen
imec-DistriNet-KU Leuven
Leuven, Belgium
firstname.lastname@cs.kuleuven.be

Elisabeth Ilie-Zudor
MTA SZTAKI, Hungarian Academy of Sciences
Budapest, Hungary
ilie@sztaki.mta.hu

Abstract

Purpose: Industry 4.0 envisions a future of networked production where interconnected machines and business processes running in the cloud will communicate with one another to optimize production and enable more efficient and sustainable individualized/mass manufacturing. However, the openness and process transparency of networked production in hyperconnected manufacturing enterprises pose severe cyber-security threats and information security challenges that need to be dealt with.

Design/methodology/approach: This paper presents a distributed trust model and middleware for collaborative and decentralized access control to guarantee data transparency, integrity, authenticity, and authorization of dataflow-oriented Industry 4.0 processes.

Findings: The results of a performance study indicate that private blockchains are capable of securing IoT-enabled dataflow-oriented networked production processes across the trust boundaries of the Industry 4.0 manufacturing enterprise.

Originality/value: This paper contributes a decentralized identity and relationship management for users, sensors, actuators, gateways and cloud services to support processes that cross the trust boundaries of the manufacturing enterprise, while offering protection against malicious adversaries gaining unauthorized access to systems, services and information.

Keywords: Interorganizational trust; Information flow; Internet; Network organization

General terms: Industrial Internet of Things; networked production; decentralized trust; blockchains

1 Introduction

The networked production of Industry 4.0 (Lee et al., 2015) and the Factory of the Future (FoF) (Karnouskos et al., 2012) will facilitate the control and optimization of individualized manufacturing where a customer's personal preferences can be swiftly fulfilled without any delay on the mass production process. However, the increased transparency of networked production poses severe cyber-security threats. The proliferation of IoT devices and CPS in networked production increases the attack surface for a malicious user to sabotage critical infrastructure, gain unauthorized access to sensitive data about customers, or intervene in collaborative production processes. These threats jeopardize the widespread adoption of Industry 4.0 processes, especially those that operate across the organizational trust boundaries of the manufacturing enterprise.

This paper presents a solution for secure and trustworthy data management based on private blockchains with capabilities for authenticating and authorizing users, things and services in IoT-enabled decentralized networked production environments, with the following key contributions:

1. Decentralized identity and relationship management for users, sensors, actuators, gateways and cloud services to support processes that cross the trust boundaries of the manufacturing enterprise (see Fig. 1).

2. A distributed trust model based on blockchains operating across the IoT and the cloud to guarantee data transparency, integrity, authenticity, and authorization of dataflow-oriented Industry 4.0 processes.

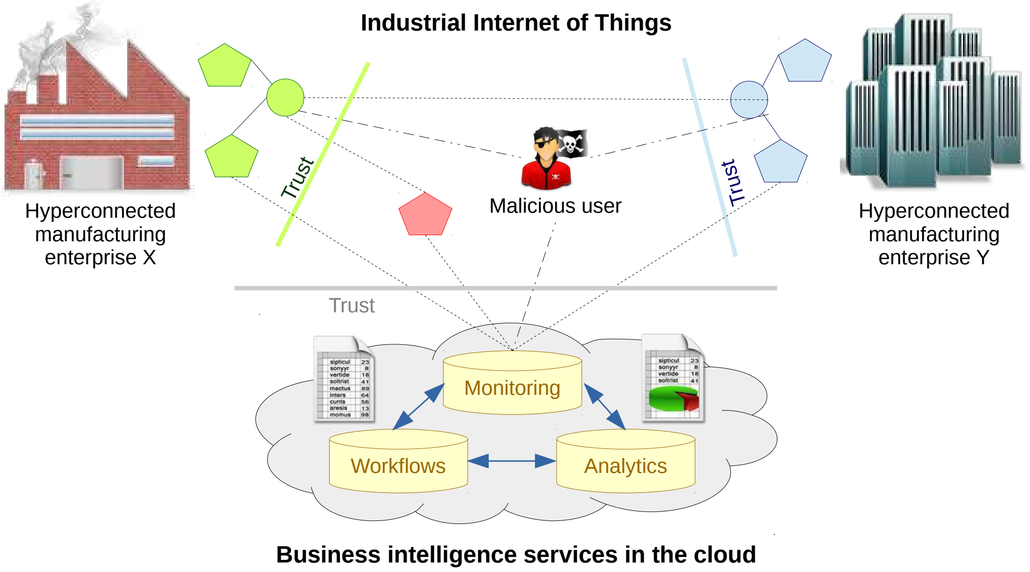
Industrial Internet of Things
Hyperconnected manufacturing enterprise X
Trust
Malicious user
Trust
Hyperconnected manufacturing enterprise Y
Trust
Monitoring
Workflows
Analytics
Business intelligence services in the cloud

Fig. 1. Networked production across organizational trust boundaries

3. Auditing support for data exchange between nodes in the production network as well as workflow compliance of these decentralized manufacturing and business processes.

The motivation for choosing blockchains (Swan, 2015) is the inherent distributed mode of operation without a centralized trust authority, as well as the increased transparency and accurate tracking benefits.

After discussing related work in section 2 and a motivating use case on networked production in section 3, we present in section 4 our blockchain-based framework for secure and trustworthy data management for IoT-enabled networked production with decentralized identity and relationship management, access control and auditing support. We evaluate our implementation and discuss performance aspects in section 5, and conclude in section 6 summarizing our main insights and identifying interesting topics for further research.

2 Background and related work

This section reviews ongoing research on key concepts of Industry 4.0, cyber-security and blockchain technology for highly resilient decentralized trustworthy services.

2.1 The Industrial Internet of Things (IIoT) and Industry 4.0

The Industrial IoT and Industry 4.0 envision improved operational efficiency in connected ecosystems of collaborating humans, machines and services (Schuh et al., 2014) by automating decisions and taking actions in real time. Herman et al. (Hermann et al., 2016) have carried out a quantitative text analysis and a qualitative literature review on Industry 4.0, identifying the main design principles behind this emerging paradigm and presenting a case study that illustrates how the principles can support practitioners in implementing Industry 4.0 scenarios. Lee et al. (Lee et al., 2015) proposed a unified 5C-level architecture (based on *connection, conversion, cyber, cognition* and *configure*) as a guideline for the implementation of CPS in the manufacturing industry. Jadzi (Jazdi, 2014) reported on the challenges regarding the development of CPS, and concerns like reliability, security and data protection. Wang et al. (Wang et al., 2016) emphasized the influence of emerging technologies such as IoT, Big Data and cloud computing in Industry 4.0, and highlighted the need for horizontal integration of inter-corporation value networks, the end-to-end integration of the engineering value chain, and the vertical integration inside the manufacturing enterprise. Monostori et al. (Monostori et al., 2016) underlined the significance of cyber-physical systems for manufacturing in Industry 4.0, benefiting from data-accessing and data-processing services available on the Internet to give rise to cyber-physical production systems. Finch (Finch, 2004) confirms that manufacturing and production systems will operate as highly interconnected and collaborating computational entities, hereby strengthening inter-organizational networking but at the same time increasing the risk exposure of supply chain enterprises. Delbufalo (Delbufalo, 2012) carried out a systematic literature review of the empirical evidence on inter-organizational trust outcomes in supply chain relationships. In line with previous research, Yang et al. (Yang and Wei, 2013) discussed the importance of developing multiple security initiatives to enhance supply chain security without jeopardizing overall efficiency. They investigated the effect of supply chain security management on security performance in container shipping operations, and their research shows that information management and partner relationship management had significant positive effects on safety performance and customs clearance performance.

2.2 Cyber-security for IoT and archetypes of production networks

Cyber-security is a non-trivial challenge and a top priority for industrial control systems (Cárdenas et al., 2008). Over the past decades, industrial control systems were deployed based on proprietary technology operating isolated from the outside world. While physical perimeter security was deemed adequate, many of the systems, technologies and non-standardized protocols were never designed with networked production in mind. These legacy systems will now be part of more open and collaborative networks, making them more vulnerable to cyber-attacks that will have a fundamentally more severe impact. Recent successful attacks on SCADA systems by dangerous malware like Stuxnet, Duqu, Flame, and Gauss (Langner, 2011; Bencsáth et al., 2012) are a mere illustration of the cyber-security challenges that IoT-enabled production networks will face in the future.

Zhu et al. (Zhu et al., 2011) highlighted the difference between SCADA systems and traditional IT systems from a security point of view and presented a taxonomy to systematically identify and classify cyber-attacks. Nicholson et al. (Nicholson et al., 2012) also presented a survey of ongoing research with an overview of risks, threats and mitigation strategies in the area of SCADA security. Similar work on cyber-security challenges in IoT was reported on in (Ning et al., 2013; Jing et al., 2014; Sicari et al., 2015).

In a report by McKinsey & Company on Industry 4.0 (McKinsey & Company, 2015), the authors argue that to address the evolution of demand three archetypes of next-generation plant models will emerge:

- **Smart automated plants**: these plants target mass manufacturing of products at a low cost and in a fully automated and digitized fashion.
- **Customer-centric plants**: these plants target mass personalization of highly customized products at scale and at an affordable cost.
- **E-plant in a box**: these plants target niche markets with small-scale mobile plants that are quickly to set up to produce a limited range of products.

Smart automated plants require full transparency of all operations to enable real-time monitoring, control and optimization of collaborative manufacturing processes, as well as predictive maintenance. When operating across the production network of the manufacturer – e.g. to share information with suppliers and distributors – further optimization improvements are possible by better matching supply and demand. However, from a cyber-security point of view, information disclosure to unauthorized parties, tampering with data and denial of service attacks are severe security threats of this increased end-to-end transparency.

For *customer-centric plants*, customers have the opportunity to influence the production process in real time to personalize the design of their products. Compared to smart automated plants where the end-user is not directly involved, confidentiality now becomes a key security requirement to to protect a customer's personal information from malicious adversaries as well as other genuine customers. Furthermore, with the upcoming EU General Data Protection Regulation (GDPR) which will be in force by May 2018, privacy compliance is an important requirement, especially for enterprises in the production network that do customer analytics and would like to sell personal data to third parties for advertising purposes.

Compared to the other archetypes, the security and privacy impact of *E-plant in a box* is likely to be smaller given the niche or remote markets they typically target. Additionally, the advantage of an e-plant in a box is that proper security and privacy measures can be fully integrated into the mobile production ecosystem.

Similar observations for the above archetypes can be made for specific production networks. For example, in the micro-electronics industry, fabless semiconductor companies will only design devices – possibly tailored towards the customer's preferences – and collaborate with a merchant foundry only in charge of manufacturing the device. Compared to an integrated device manufacturer (IDM) that handles both the design and manufacturing in-house, the former must address more stringent information disclosure concerns when enabling end-to-end process transparency across the organizational boundaries of the enterprise.

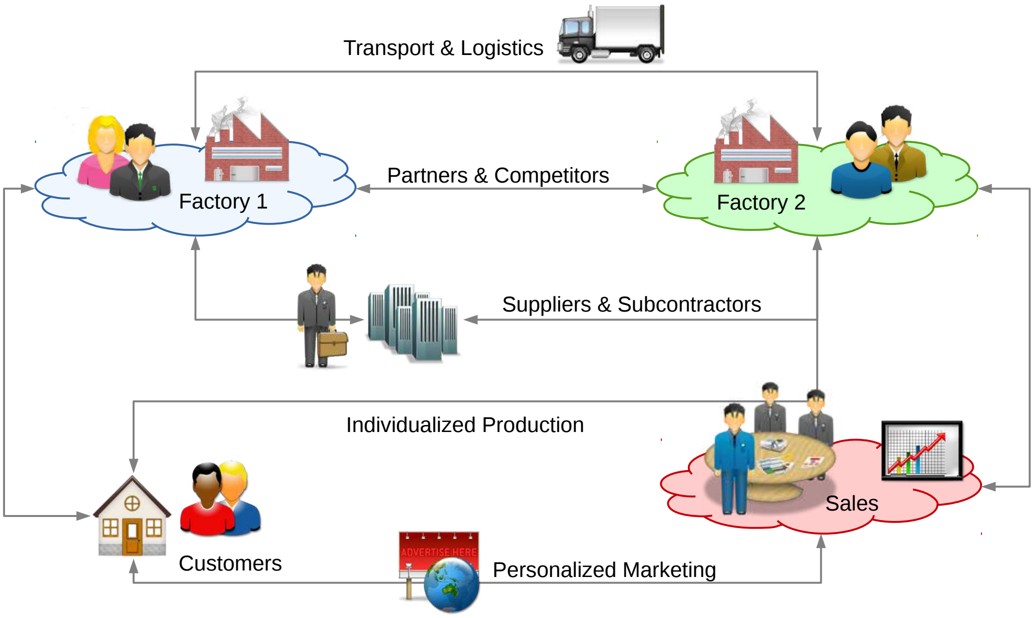

Fig. 2. Networked production and concerted manufacturing processes

2.3 Distributed trust with decentralized blockchains

The distributed nature of the IoT and the continuously evolving interactions between business stakeholders in production and logistic processes (as competitors or partners) make Industry 4.0 a highly decentralized ecosystem in which it is highly unlikely that a single authority can deliver or enforce trust among all users, things and services in the production network. The lack of a centralized trust authority was the main motivation behind the construction of blockchains (Swan, 2015) with the Bitcoin *cryptocurrency* (Nakamoto, 2008; Tschorsch and Scheuermann, 2016; Eyal et al., 2016) being the most popular application of this innovative technology.

A Bitcoin blockchain is inherently a distributed ledger with Byzantine fault-tolerant (Castro and Liskov, 1999) consensus, i.e. a highly resilient peer-to-peer database architecture maintaining blocks of transactions that each contain a timestamp and a reference to a previous block (hence the name *blockchain*). Each transaction is hardened against tampering and revision by applying cryptographic digital signatures. A new transaction is broadcasted to the P2P network, and the digital signatures of the transaction will be verified by a network of *miners*. Once confirmed, the transaction is combined with other transactions into a new block that is permanently added to the blockchain.

Beyond financial applications, blockchains have been explored for implementing *smart contracts* (Kosba et al., 2015) to run arbitrary user-defined programs on the blockchain, to protect personal data or guarantee anonymity (Zyskind et al., 2015; Miers et al., 2013).

2.4 Interoperability with industrial standards

A key advantage of the OPC Unified Architecture (OPC UA) platform-independent standard (IEC 62541) for machine-to-machine communication between industrial automation devices and systems, and the AutomationML specification (IEC 62714) for XML-based interoperability of online production data, is that it reduces the effort required to integrate cross-domain systems and exchange information between factories.

The OPC UA standard specifies security profiles for authentication, authorization and encryption. Upon connecting, a user must authenticate himself with a username/password, a X.509 certificate, or via Kerberos. OPC UA applications identify themselves in a similar way with software certificates. Access rights to data can be specified in a fine-grained manner. The message integrity is guaranteed with digital signatures, whereas confidentiality of transmitted user data is achieved through OpenSSL encryption.

The *Secure Plug and Work* research project of Fraunhofer IOSB[1] is one initiative that adopted these open specifications to enable the plug-and-work capability in production-oriented software components. Within the

[1] `http://www.secureplugandwork.de`

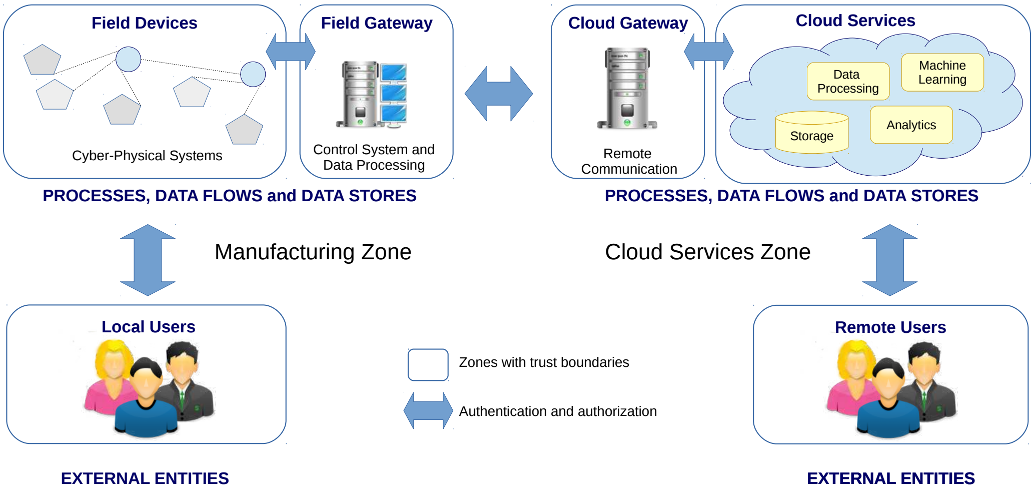

Fig. 3. Zones with trust boundaries, and authenticity of identity in networked production

frame of this project, Schleipen et al. (Schleipen et al., 2015) proposed a role-based authorization scheme to provide limited access to data on a user level. We will refer back to this work in Section 5 when we compare the security advantages and limitations of our blockchain-based framework.

3 Motivating scenario on IoT-enabled networked production

In networked production, people, machines and business processes will interact with one another (see Fig. 2) to enable personalized products through flexible, resource-efficient and cost-effective manufacturing. Interconnected systems will be linked to cloud services for remote monitoring and data analytics to optimize production plans, enable proactive maintenance, and respond quicker to changing customer requirements.

The security and privacy threats we consider in this motivating scenario focus mainly on the customer-centric plants, as it is the most challenging one with respect to two well-known threat models (the honest-but-curious and the malicious adversarial threat model). To assess all threats we model the networked production workflow as a dataflow diagram on which we carry out a STRIDE security analysis (Scandariato et al., 2015). In Fig. 3 we illustrate the different trust zones in networked production. We identify the following core elements in the threat model:

- **Processes**: field devices, field gateway and cloud services
- **Communication**: between all field devices, the field gateways and cloud gateways
- **Storage**: field devices and field gateway

To protect against spoofing attacks where devices are replaced with malicious ones, or escalation of privilege threats where compromised nodes would allow operations that are otherwise not possible, we need to put authentication and authorization mechanisms in place. The communication between nodes in the production network should be protected against eavesdropping or data interference by encrypting all traffic as a countermeasure. This mitigates the tampering, information disclosure, and denial of service threats. All data is stored encrypted, and digital signatures protect the data against repudiation attacks.

4 Secure and trustworthy data management with private blockchains

To ensure secure and trustworthy data management, all nodes build upon a trust model that provides the following guarantees as key security requirements:

1. The identity of each node – i.e. a user, thing or service – in the production network can be verified by every other node in the network.
2. Every data exchange by a node can be verified in a decentralized manner.

3. Unauthorized nodes do not have access to data to preserve the confidentiality of sensitive information.
4. Access privileges can be revoked by the data owner in response to changing relationships with other nodes.

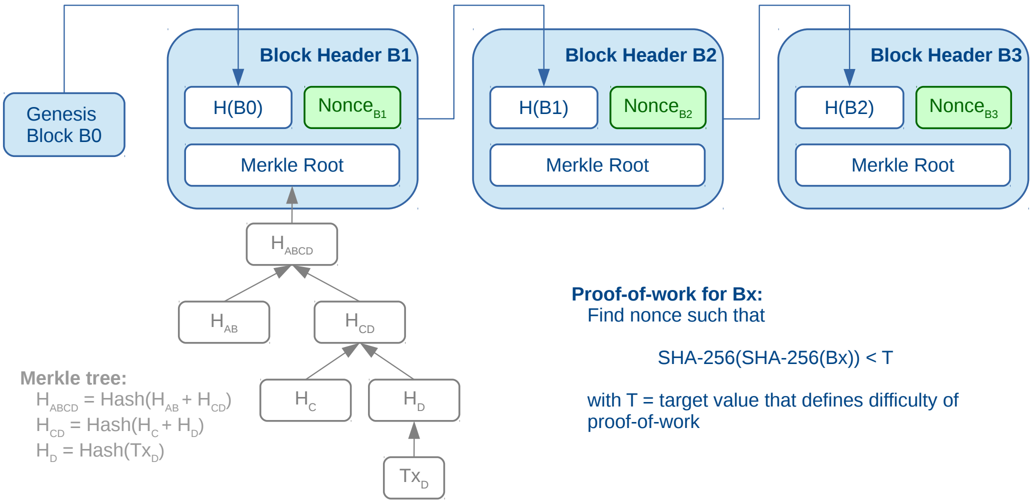

Fig. 4. Chained blocks of transactions are approved by miners delivering a proof-of-work

4.1 Trustworthy data storage in a decentralized private blockchain

Blockchains are a permanent, immutable, highly resilient and decentralized P2P data store that offer specific security and trust guarantees for connected clients. In the case of Bitcoin (Nakamoto, 2008), clients can trust transactions on the blockchain, even without a centralized trust authority that safeguards them against other fraudulent clients. A user may inadvertently or maliciously try to spend the same Bitcoins twice before the first transaction is confirmed, and this may lead to a double spending fraud case. An adversary could initiate two transactions with the same Bitcoins – one to a genuine receiving party and another to himself – and it would not be clear which of these competing transactions would win, leading to a possible disadvantage for the receiving party if he delivered the goods or services without being rewarded. To mitigate such double spending attacks, miners that approve an unconfirmed block of transactions, must deliver a *proof-of-work* (see Fig. 4). This proof-of-work is a computationally expensive process, i.e. finding a value for the *nonce* in the block header such that a double SHA-256 hash on the block header (containing this nonce, the hash of the previous block, the hash of the root of the Merkle tree (Merkle, 1988) on the transactions in the block, and a few other fields) is below a given target value. As a financial incentive to participate in the approval process, the first one among many competing miners to deliver the nonce as proof-of-work receives 25 Bitcoins as well as the transaction fees associated with the transactions in that block.

These are interesting trust properties for the Industry 4.0 collaborative business process and manufacturing scenarios. However, Bitcoin only supports financial transactions between addresses as the single type of asset in the blockchain. While it is technically feasible to use the public Bitcoin blockchain to store the above profile and other information as extra metadata in a Bitcoin transaction, the distributed immutable ledger of Bitcoin was never meant to serve as a general purpose decentralized data store. Other disadvantages of using the public Bitcoin blockchain are (a) the transaction fee of each transaction as an incentive for miners to verify the transaction, (b) the slow registration of a block of transactions in the blockchain (about 1 block every 10 minutes), (c) the fact that the blockchain currently already holds more than 92GB of data[2] that is irrelevant for our application domain, and (d) the lack of any permission system to control who can connect to and operate on the blockchain. These are serious constraints that jeopardize the practical and technical feasibility of our Industry 4.0 application. Rather than using Bitcoin or an alternative public blockchain like Ethereum[3] and Hyperledger[4], our proof-of-concept relies on a private blockchain for increased flexibility and control.

[2] https://blockchain.info/charts/blocks-size
[3] https://www.ethereum.org
[4] https://www.hyperledger.org

4.2 Trustworthy digital identities and profiles

Every user, thing (i.e. sensor, actuator, mobile device, production machinery) and cloud service has a digital identity based on a cryptographic keypair (i.e. a public and private key). This identity is registered on the distributed blockchain which has the advantage that identity provisioning and verification of the uniqueness of identities can be achieved without a global centralized overseeing identity providing authority. Furthermore, in a decentralized world like the IoT, device-to-device authentication can take place without any intermediaries acting as a trusted third party to confirm the identity of a device.

A cryptographic keypair is generated according to the *secp256k1* Elliptic Curve Digital Signature Algorithm (ECDSA) standard. The OpenSSL command in Listing 1 illustrates how to generate a private key *priv-ec256.pem* that is 256 bits or 32 bytes long.

```
$ openssl ecparam -genkey -name secp256k1 -out priv-ec256.pem

$ openssl ec -in priv-ec256.pem -text -noout
read EC key
Private-Key: (256 bit)
priv:
    75:3c:59:9e:d0:1c:c2:92:94:dc:de:55:64:ff:b6:
    a1:9d:08:14:e5:2c:21:0d:bb:09:e5:ab:31:89:37:
    69:3a
pub:
    04:a2:c6:2a:f2:e5:1c:6e:31:a9:6e:e6:7a:86:18:
    93:c3:4e:90:31:ae:e9:34:46:65:82:1d:5c:b9:99:
    90:52:20:10:a4:29:cc:f0:fa:87:71:e2:a6:c1:41:
    cc:bd:7b:6d:7c:bd:68:ec:d5:06:9f:17:93:b0:02:
    c6:10:fd:e5:39
ASN1 OID: secp256k1
```

Listing 1. Generate and print a private key of an EC256 keypair

The public key of the cryptographic keypair is derived from the private key, and is made up of two numbers of each 32 bytes long that respectively represent the x and y coordinates on the *secp256k1* elliptic curve:

$$y^2 \bmod p = (x^3 + 7) \bmod p \qquad \text{with } p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1 \tag{1}$$

The actual x, y location on the curve is defined by the private key, but inferring the latter based on the two coordinates is unfeasible. The OpenSSL command to create the public key *publ-ec256.pem* of the Elliptic Curve (EC) cryptographic keypair is shown in Listing 2.

```
$ openssl ec -in priv-ec256.pem -pubout -out publ-ec256.pem
read EC key
writing EC key

$ openssl ec -in publ-ec256.pem -pubin -text -noout
read EC key
Private-Key: (256 bit)
pub:
    04:a2:c6:2a:f2:e5:1c:6e:31:a9:6e:e6:7a:86:18:
    93:c3:4e:90:31:ae:e9:34:46:65:82:1d:5c:b9:99:
    90:52:20:10:a4:29:cc:f0:fa:87:71:e2:a6:c1:41:
    cc:bd:7b:6d:7c:bd:68:ec:d5:06:9f:17:93:b0:02:
    c6:10:fd:e5:39
ASN1 OID: secp256k1

$ openssl ec -in publ-ec256.pem -pubin -text -noout -conv_form compressed
read EC key
Private-Key: (256 bit)
pub:
    03:a2:c6:2a:f2:e5:1c:6e:31:a9:6e:e6:7a:86:18:
    93:c3:4e:90:31:ae:e9:34:46:65:82:1d:5c:b9:99:
    90:52:20
ASN1 OID: secp256k1
```

Listing 2. Derive and print the public key in uncompressed and compressed format

As the y coordinate depends on the x value, the public key can also be stored in compressed format. One can distinguish between both by their length and the first byte (either 0x04 or 0x03).

Each identity can be associated with a profile in which arbitrary information about the node can be stored. To ensure interoperability, the data is represented in a JSON schema as depicted in Listing 3 that gives an example of a user profile for the user *john*.

```json
{
  "id":"john",
  "givenName":"John",
  "sn":"Doe",
  "role": [ "admin", "miner" ],
  "trustZone":"Factory1",
  "mail":"john.doe@company.com"
}
```

Listing 3. A user profile in JSON format

User *john* digitally signs his user profile using his private key *privkey*:

$$\sigma = \text{sign}(\text{privkey, profile}) \tag{2}$$

The public key *pubkey*, the digital signature σ, and the profile are broadcasted as metadata on the blockchain. The blockchain miners in the production network can verify that (a) the identity has not been registered before in the blockchain, and (b) the digital signature σ on the identity and profile corresponds with the public key *pubkey* of the user. This way, they can ascertain the ownership of the corresponding private key by the user, and link the digitally signed information stored in the user profile with the cryptographic keypair.

4.3 Keypair updates and revocation within a trust zone

The identity of each node in the production network is associated with a public and private key. The production network must reliably deal with situations where the cryptographic keypair was created with a limited lifetime, or where an adversary has compromised the private key of a node on the network. Under these conditions, the keypair can no longer be trusted.

If the keypair will expire, the node can still create a new keypair, and sign it off with the old private key.

$$\sigma_1 = \text{sign}(\text{privkey}_{old}, \text{ id} \mid \text{pubkey}_{new}) \tag{3}$$

$$\sigma_2 = \text{sign}(\text{privkey}_{new}, \text{ id}) \tag{4}$$

In the case a malicious adversary obtained access to the private key, the keypair must be revoked. The legitimate owner can still issue a digitally signed statement to revoke the cryptographic keypair, so that all future transactions by the adversary will not be confirmed by the blockchain miners.

An alternative is to update the compromised keypair, using a similar mechanism as proposed in CertCoin (Conner Fromknecht, 2014) where another second offline identity (and associated cryptographic keypair) is used to update the compromised keypair with similar statements as above but now digitally signed with *multiple* private keys. The idea behind the scheme is that an adversary does not have access to the private keys of the additional identities. Such a multi-party signed update statement would override a single-party signed update statement by an adversary that can only sign with the compromised private key. In principle, the second offline identity can belong to the same node. However, in our Industry 4.0 blockchain solution, this identity should belong to a human being (rather than an autonomous machine, device or other thing) that is a member of a designated set of collaborative administrators responsible for the trust zone of the owner of the compromised keypair. In both cases, the update or revocation statement is broadcasted on the blockchain and independently verifiable by other miners on the decentralized peer-to-peer network.

4.4 Network of trust in digital identities

The profile of an entity can store links to other identities of nodes in the production network as a way to establish a network of trust. Rather than storing the trust information directly in the profile, our framework allows a node to vouch for the identity and profile of another node by digitally signing such a *vouch* statement (see Listing 4), and storing it as metadata of a zero amount transaction to itself in the immutable decentralized blockchain.

```
{
  "vouch":<address>
}
```

Listing 4. Vouching for another trusted identity as transaction metadata

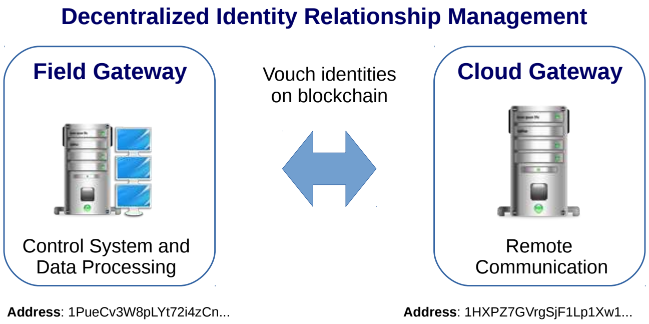
Decentralized Identity Relationship Management
Field Gateway
Control System and Data Processing
Vouch identities on blockchain
Cloud Gateway
Remote Communication
Address: 1PueCv3W8pLYt72i4zCn...
Address: 1HXPZ7GVrgSjF1Lp1Xw1...

Fig. 5. Transitive trust relationships based on vouch statements on the blockchain

The address in the above metadata is made public by the trusted user himself. It is computed as the RIPEMD160/SHA256 double hash of the public key *pubkey* of the user, subsequently prepended with 0x00 (address on the main Bitcoin network) and then Base58Check-encoded, as demonstrated in Listing 5:

$$\text{address} = \text{Base58Check}(\text{0x00} \mid\mid \text{RIPEMD160}(\text{SHA256}(\text{pubkey})))$$

```
pubkey: 03a2c62af2e51c6e31a96ee67a861893c34e9031aee9344665821d5cb999905220
double hash: b54151744821f707150055b32ef8a5834c1b3422
address: 1HXPZ7GVrgSjF1Lp1Xw1riYrAbMFHhZq5N

{
  "vouch":"1HXPZ7GVrgSjF1Lp1Xw1riYrAbMFHhZq5N"
}
```

Listing 5. Compute the address based on the compressed public key

Collectively, these vouch statements are the foundation for our decentralized identity relationship management in which a network of trust between digital identities is established. The transitive trust relationships are a means to bootstrap interaction between previously untrusting nodes in collaborative but decentralized Industry 4.0 business processes.

4.5 Access control for decentralized workflows

Nodes in the production network exchange data with one another, either directly or through the underlying peer-to-peer architecture of the decentralized blockchain. We now discuss in more detail how nodes establish trustworthy communication between one another.

4.5.1 Authentication of communicating peers in the production network

The output of one process of a node in the IoT-enabled production network serves as input for the next task in the workflow. In order to mitigate spoofing or man-in-the-middle attacks, the authenticity of the identity of the sending and receiving party must be established to avoid unintended information disclosure. This is achieved through the following simplified high-level mutual authentication scheme:

- Source s sends a nonce ns to target t
- Target t sends a nonce nt and the signature $st = sign_t(nt||ns)$ to source s
- Source s verifies the signature st with the public key of target t
- Source s sends the signature $ss = sign_s(ns||nt)$ to target t
- Target t verifies the signature ss with the public key of source s

The random nonces can only be used once, and are meant to prevent chosen plaintext attacks by a malicious adversary. Additionally, each party must have access to the public key of the counterparty and verify its validity against the blockchain. If both parties have shared a common secret s in advance, the mutual authentication can be implemented by computing and comparing cryptographic hash functions rather than by verifying digital signatures, and this by respectively computing $ht = hash(nt||ns||s)$ and $hs = hash(ns||nt||s)$ in the above scheme instead. This avoids the expensive traversal of the whole blockchain to collect and validate the public key of the counterparty.

4.5.2 Privileges and policy-based authorization

Public blockchains like Bitcoin are open for everyone to connect, and every member has the opportunity to send or receive transactions or validate transactions as a miner. To safeguard the integrity and confidentiality of sensitive data, our framework aims for more tighter controlled permissions on who can connect, send and receive data, act with administrative capabilities, vouch for other nodes to establish a network of trust, mine blocks of transactions, etc. For these purposes, every node must acquire the necessary privileges before it can assume a role in the IoT-enabled production network.

We previously outlined how mutual authentication is used to establish the true identity of nodes exchanging data. To meet the security requirements discussed earlier, each node must also ascertain under which terms and conditions the counterparty is authorized to send or receive data. Each node can issue a request to a designated policy decision node (e.g. the gateway within a trust zone) to evaluate its authorization policies against a given exchange of data. These policies are defined as a set of rules and conditions on attributes defined in the profile of the node, the trust level, etc. Our framework adopts a lightweight policy language also suitable for resource constrained devices.

```
if (source.role.contains("gateway") && target.role.contains("gateway")) {
   return "permit";
}

if (source.role.contains("admin")) {
   return "permit";
}

if (!source.role.contains("write")) {
   return "deny";
}

if (!target.role.contains("read")) {
   return "deny";
}

if (source.trust(target)) {
   return "permit";
}

if (source.trustZone != target.trustZone) {
   return "deny";
}

return "permit";
```

Listing 6. Policy-based authorization

Listing 6 gives a simple example of policy-based authorization depending on the roles and trust zones of the nodes involved in the data exchange, as well as if there is an indirect trust relationship between both identities (as discussed in a previous section). Both the source and target in the above authorization policy are represented by their blockchain *address*. The policy is signed with the offline private key of the policy decision node to circumvent unauthorized modifications of the policy.

4.5.3 Auditing the data flow of process orchestrations in the production network

To verify the accountability of all nodes involved in a data flow, all transactions between a source and target node are registered in the blockchain to ensure non-repudiation. To make the actual exchange auditable in the future, both nodes participate in the following communication protocol:

- Source s sends a salted hash of the data $hs = hash(salt||data)$ and the salt $salt$ to target t as metadata on the blockchain.
- Target t sends a digitally signed request (containing the above salt and hash of the data) to source s.
- Source s sends the data encrypted and digitally signed to target t.
- Target t verifies the identity of the source and compares the data against the hash in the first transaction.
- Target t sends an acknowledgment transaction on the blockchain with the original hash as metadata.

The above approach guarantees that both the source and target can prove that a particular message was sent and received.

In order to audit a decentralized data flow, i.e. trace the nodes that produced and consumed data, we compute the hash not only on the data being exchanged between the source and target, but also on the hashes of the data dependencies that served as input. Each output of a processing task holds a reference to the outputs of previous tasks or sensors upon which it depends, such that not only the authenticity of the identity of the node handling the data can be verified, but also the integrity of the overall data flow. Note that the actual data exchange between nodes in the production network is not registered on the blockchain (only the hash is), but takes place through encrypted communication between the nodes.

4.6 Implementation

Our framework builds on top of state-of-practice components, one of which provides the basic foundations for a private blockchain. We carried out a technical feasibility analysis of the following blockchain frameworks:

- Ethereum: `https://www.ethereum.org`
- MultiChain: `http://www.multichain.com`
- Hyper Ledger: `https://www.hyperledger.org`
- Openchain: `https://www.openchain.org`

A selection was made based on (1) the ability to set up a private blockchain, (2) complexity of deployment, (3) availability of security features, and (4) interoperability to easily integrate missing capabilities. Ethereum is basically a public blockchain, but a private one can be set up as a test net that is separate from the main Ethereum chain. Openchain is not really a blockchain by definition, as it misses the concept of a block and rather chains transactions together. Furthermore, setting up Hyper Ledger or Openchain as a docker image prohibits deployment of the blockchain client on resource constrained devices. In that sense is MultiChain more lightweight and far less complex to deploy and configure, while offering most of the basic blockchain functionalities that our secure and trustworthy data management framework requires.

The additional functionality is implemented as a microservice using the Spring Boot[5] framework. The latter is deployed on all nodes in the production network and interacts with the private MultiChain peer-to-peer network with JSON-RPC API commands. The functionality that the microservice adds, includes:

1. Provision identity profiles and trust zones for the production network
2. Manage a network of trust across organizational boundaries
3. Keypair revocation and multi-party signed keypair updates
4. Augmented permission model and policy-based authorization
5. Auditing of data exchange and distributed workflow compliance

The main motivation of why the above capabilities were implemented as separate features in a microservice, and not directly into Multichain were twofold: (1) decouple the microservice from the Multichain daemon for very resource constrained devices, and (2) simplify the transition to and interoperability with other blockchain technologies. Also, an ARM port of the Multichain blockchain was not available for deployment on our ODROID-U3 mini-boards. The proof-of-concept was therefore evaluated on a cluster of 64-bit Ubuntu 15.10 Linux nodes.

5 Evaluation

In this section, we evaluate our solution in a proof-of-concept Industry 4.0 hyperconnected production network scenario in which we define the following trust zones:

- 3 *manufacturing* trust zones with each 50 monitoring nodes for diagnostic activities and 10 data processing nodes to control the individualized production and manufacturing.

[5] `http://projects.spring.io/spring-boot/`

- 5 *logistics* trust zones with each 10 nodes to track the packaging, and trace shipment and delivery of customer orders. Each transportation vehicle acts as a monitoring node.
- 2 *analytics* trust zones deployed with 3 nodes running business processes in the cloud. The 3 service nodes are in charge of data analytics for sales forecasts and monitoring of excess or obsolete products for which sales teams can offer discounts in targeted marketing campaigns.

Each of the trust zones has dedicated administrative nodes, and gateway nodes to interact and establish trust relationships with other trust zones. In total, our experimental setup for IoT-enabled networked production has 240 nodes. We use an experimental setup of 15 machines, each equipped with an Intel Core 2 Duo 3.00 GHz CPU and 4GB of memory and running a 64-bit Ubuntu 15.10 operating system. Each machine runs 16 Multichain daemons to represent the production network of 240 nodes. All machines are linked to a 1 Gigabit network. The microservice runs on each machine on top of the 64-bit Java SE Development Kit version 8u101.

For the evaluation, we will evaluate the performance and latency implications of the key features of our secure and trustworthy data management framework, and analyze it from a security point of view with respect to the requirements outlined earlier.

5.1 Performance and practical feasibility

In the experimental setup, we do not measure the performance of the core business logic of the service or the impact of direct (encrypted) communication between nodes in the production network, but only the computational overhead of using our framework on top of the Multichain private blockchain implementation. More specifically, we measure the latency of:

1. Time to provision identity profile
2. Time to authenticate a counterparty
3. Time to update cryptographic keypair with multi-party signature
4. Time to audit node-to-node data exchange and overall workflow

For the transactions that take on the blockchain, we keep the default setting of Multichain of a target time between blocks of 15 seconds.

5.1.1 Provisioning identity profiles on the blockchain

In Fig. 6 we illustrate the latency on block confirmations for respectively 1 and 8 miners, and this for 5 different runs of an identity profile being registered on the blockchain. Note that the target time of 15 seconds between every block confirmation is merely a rough estimate and that adding more miners will reduce the overall latency.

5.1.2 Latency on mutual authentication

In a previous section, we discussed the mutual authentication protocol for 2 nodes in the production network that want to exchange data. In our setup, the nodes are able to communicate directly with one another, and make use of Multichain's APIs to sign and verify challenge-response messages:

```
signmessage "1HXPZ7GVrgSjF1Lp1Xw1riYrAbMFHhZq5N" "123,456"
verifymessage "1HXPZ7GVrgSjF1Lp1Xw1riYrAbMFHhZq5N" "H3kO9Zt..." "123,456"
```

Listing 7. Signing and verifying challenge-response messages

In Listing 7, the parameter *1HXPZ7G...* is the address of the signing party, and "123,456" represents the random nonces of the source and target nodes. The message verification is carried out by the receiving party, with *H3kO9Zt...* being the Base64-encoded digital signature as result from the first step. Depending on the connectivity and bandwidth of the communication channel between the nodes involved, the mutual authentication in our experimental setup took less than 370 msec.

Note that this mutual authentication interaction is not registered on the blockchain. However, each node must ascertain the validity of the cryptographic keypair of the counterparty and traverse the blockchain to verify whether the keypair has been updated with a new one, possibly after a revocation due to a compromised. When a node has not interacted with the counterparty before, it leverages Multichain's data stream API to subscribe

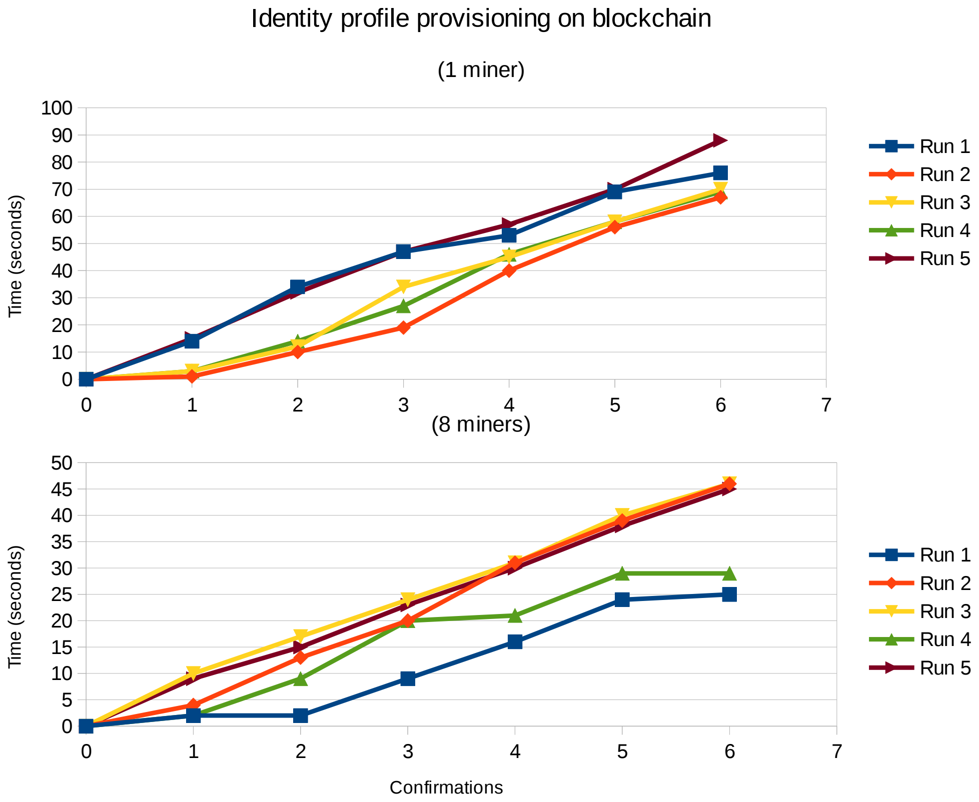

Fig. 6. Latency on block confirmations for identity provisioning with 1 and 8 miners

to the address of the counterparty. At this point the blockchain is rescanned for revocation transactions sent to this address. In our experimental setup, this took less than 10 seconds. However, depending on the actual size of the private blockchain, this can take considerable more time.

5.1.3 Revoking and updating a keypair with signatures of multiple parties

In terms of complexity, this experiment on the blockchain combines the latencies of the 2 previous experiments. Issuing a revocation statement is similar to provisioning an identity profile on the blockchain. As such, the performance results on the latency of blockchain confirmations are comparable to those reported in Fig. 6.

When multiple parties must sign a keypair update transaction with their *offline* identity, each of the corresponding keypairs must be checked for validity as well, leading to similar latency concerns as in the previous experiment. As the offline identities for each trust zone that can sign such a multi-party keypair update transaction request are known upfront in our Industry 4.0 production network ecosystem, we minimize the latency by subscribing each node upfront to transactions of all offline identities in the same trust zone. The multi-party keypair update transaction using 3 offline identities takes about 1 to 3 seconds in our setup, depending on the load and network traffic of the signing nodes. After registering the update transaction on the blockchain, the miners on the blockchain can start validating the transaction and the other nodes will have 6 block confirmations after about 1 minute.

5.1.4 Auditing the node-to-node communication and the overall dataflow

For this experiment, we must distinguish between ascertaining the authenticity of the nodes and the integrity of the data being exchanged *during* the interaction in realtime, and *after* the transaction has been completed as a post-factum security assessment. A similar observation can be made about auditing the overall end-to-end dataflow of a decentralized workflow. For practical purposes, it is not realistic for a target node to wait for 6 block confirmations on the hash of the data that a source node registered on the block chain before continuing his part of the workflow.

We simulated a workflow of up to 1000 message exchanges between 30 nodes. Each transaction registered on

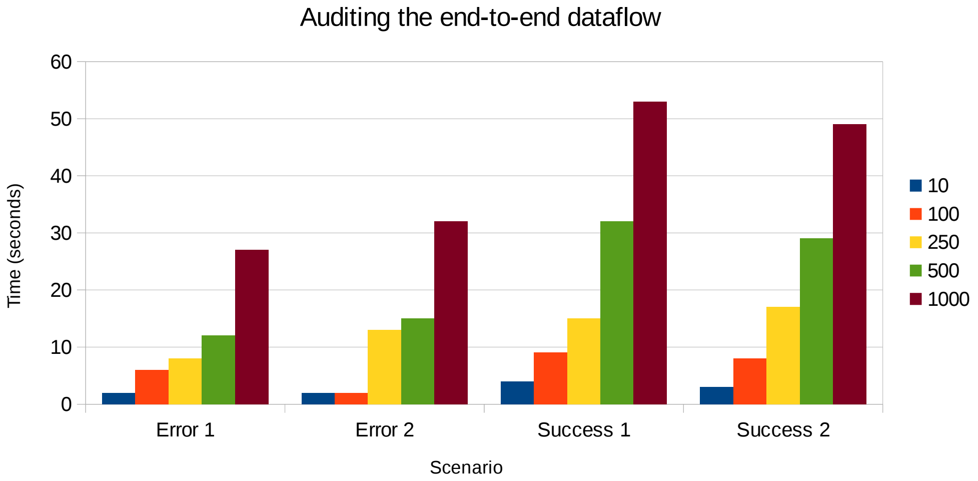

Fig. 7. Auditing the transactions on the blockchain for growing process workflows

the blockchain relies on 0 (for sensors and monitors) to 10 inputs. We measured the time it takes to verify that (1) each data exchange has been validated by 6 block confirmations, (2) the overall data flow is consistent with the authorization policy and workflow. Under the assumption that the node carrying out the audit has local access to the whole blockchain, the compliance verification was completed under a minute in case of success, and faster when an (enforced) error had occurred. Fig. 7 illustrates the time it takes in terms of the amount of messages exchanges in the workflow for two success and two error scenarios.

5.2 Security and privacy analysis

In this section, we will revisit the security requirements outlined earlier and briefly outlined the consequences in terms of an *honest-but-curious* adversary model and an attack model with malicious users.

- *Spoofing:* In order for a spoofing attack to be successful, a malicious user must have gained access to the private key of a node on the production network. Such an attack can be thwarted by having the private key stored in the Trusted Execution Environment (a secure area of the main processor).
- *Tampering:* The blockchain ensures by design that a malicious user cannot tamper with the transactions stored in the blockchain.
- *Repudiation:* A node cannot repudiate the authenticity of a transaction on the blockchain as they are all signed with his private key.
- *Information disclosure:* Beyond the profile of each node, our framework does not store any sensitive information on the blockchain. An honest-but-curious node can observe the transactions and the hash of the data exchanged between certain nodes.
- *Denial-of-service:* The blockchain is an inherently decentralized system with redundancy built-in, such that it does not exhibit a single point of failure. The peer-to-peer nature of blockchains makes them more resilient against denial-of-service attacks, allowing for an increased availability.
- *Elevation of privilege:* An attacker may compromise a node and gain access to local resources. However, the enforcement of policy-based access control prohibits him from gaining unauthorized access to information from other nodes, unless he is able to compromise the offline private key of the policy decision node to modify and sign the access policy.

All data concerning the manufacturing processes and business logic of processes in the cloud is communicated encrypted outside the blockchain. While this offers confidentiality of all interactions, it is not sufficient to guarantee anonymity on the blockchain. While it is not possible to derive the public key of a node based on its blockchain address, it is revealed if an attacker has been involved in a previous genuine transaction with its victim. This makes identifiability and linkability of transactions two privacy threats that cannot be mitigated with our solution.

Security capabilities	**OPC UA**	**Blockchain-based architecture**
Authentication	✓	✓
Authorization	RBAC	ABAC
Trust authority	Centralized	Decentralized
Non-repudiation	✓	✓
Confidentiality	✓	✓
Customer access control	✗	✓
Tamper protection	✗	✓
Auditing transparency	✗	✓
Availability (DoS protection)	✗	✓
Anonimity and Privacy	✗	✗

Table 1: Security comparison of an OPC UA instantiation with the proposed blockchain-based architecture

5.3 Benefits and disadvantages

Based on previous work (Huang et al., 2010; Schleipen et al., 2015; Wu et al., 2015; Jung et al., 2017), we compare in Table 1 the security features and capabilities of a typical OPC UA architecture with our blockchain-based architecture. The added value of our blockchain based implementation is that it allows for greater transparency and auditing of all processes for the customer, whereas access by external parties to machines and devices of a factory is typically restricted. Beyond the more sophisticated Attribute-Based Access Control (ABAC) which allows for more flexibility in expressing authorization policies, our solution offers better auditing capabilities and availability. OPC UA typically provides traceable log entries for auditing, which an adversary can easily tamper with, and presumes availability by minimizing any processing before the authentication. However, this means that authentication capability of an OPC UA server itself may be subject to a Denial of Service (DoS) attack and possibly become a single point of failure. The attestation of events and the decentralized design make our blockchain-based architecture more resilient against these security concerns.

While the proposed blockchain-based solution has clear benefits for customer-centric plants, there are also disadvantages regarding cost and managerial overhead for the other archetypes of next-generation plant models. Indeed, smart automated plants do not require customer access control to deliver highly customized products, and an E-plant in a box may be targeting such a small niche market with a limited range of products that setting up the block chain is too time demanding.

6 Conclusion

Linking production facilities to the Internet and connecting them to the cloud for remote monitoring and data analysis opens them up to severe security threats, ranging from sabotaging critical infrastructure from the outside, to unauthorized access to sensitive customer data and industrial espionage from within. Contemporary solutions exist to address these concerns on a small scale or in closed loop deployments with trusted third parties, they are inadequate for large scale collaborative customer-centric production networks where a centralized authority is not available. The adoption potential of Industry 4.0 and its ambition towards full end-to-end transparency will not be reached if these threats cannot be mitigated.

Based on a security threat assessment of a typical production network, we presented a solution for trustworthy networked production based on private blockchains with the ability authenticate and authorize users, things and services, and audit their interactions, all in a decentralized manner. This way, our solution can mitigate identity spoofing, information disclosure and escalation of privilege threats, both with as well as across the trust boundaries of the Industry 4.0 manufacturing enterprise.

Two important challenges remain unaddressed in this work. The first relates to the growing size of the private blockchain. The size of public Bitcoin blockchain is already more than 92GB. Even the size of our private blockchain would be 100 times smaller, size may still be a concern for resource constrained devices, prohibiting them from mining (or verifying) all transactions. The second challenges deals with privacy. Our solution cannot avoid identifiability and linkability attacks that may harm the privacy of the customer. Our current STRIDE security assessment should be complemented with the LINDDUN privacy assessment based on the same dataflow model. Beyond identifying additional privacy threats, LINDDUN can propose relevant privacy enhancing techniques. For example, mathematical protocols like zero knowledge proofs could be a complementary solution to minimize information disclosure at the expense of being computationally much more expensive. The feasibility of applying additional privacy enhancing techniques on top of private blockchains will be part of future work.

References

Bencsáth, B., Pék, G., Buttyán, L. and Félegyházi, M. (2012), 'The cousins of stuxnet: Duqu, flame, and gauss', *Future Internet* **4**(4), 971.

Cárdenas, A. A., Amin, S. and Sastry, S. (2008), Research challenges for the security of control systems, *in* 'Proceedings of the 3rd Conference on Hot Topics in Security', HOTSEC'08, USENIX Association, Berkeley, CA, USA, pp. 6:1–6:6.

Castro, M. and Liskov, B. (1999), Practical byzantine fault tolerance, *in* 'Proceedings of the Third Symposium on Operating Systems Design and Implementation', OSDI '99, USENIX Association, Berkeley, CA, USA, pp. 173–186.

Conner Fromknecht, Dragos Velicanu, S. Y. (2014), 'A decentralized public key infrastructure with identity retention', Cryptology ePrint Archive, Report 2014/803.

Delbufalo, E. (2012), 'Outcomes of interorganizational trust in supply chain relationships: A systematic literature review and a metaanalysis of the empirical evidence', *Supply Chain Management: An International Journal* **17**(4), 377–402.

Eyal, I., Gencer, A. E., Sirer, E. G. and Renesse, R. V. (2016), Bitcoin-ng: A scalable blockchain protocol, *in* '13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)', USENIX Association, Santa Clara, CA, pp. 45–59.

Finch, P. (2004), 'Supply chain risk management', *Supply Chain Management: An International Journal* **9**(2), 183–196.

Hermann, M., Pentek, T. and Otto, B. (2016), Design principles for industrie 4.0 scenarios, *in* 'HICSS', IEEE Computer Society, pp. 3928–3937.

Huang, R., Liu, F. and Dongbo, P. (2010), Research on opc ua security, *in* '2010 5th IEEE Conference on Industrial Electronics and Applications', pp. 1439–1444.

Jazdi, N. (2014), Cyber physical systems in the context of industry 4.0, *in* 'Automation, Quality and Testing, Robotics, 2014 IEEE International Conference on', pp. 1–4.

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J. and Qiu, D. (2014), 'Security of the internet of things: perspectives and challenges', *Wireless Networks* **20**(8), 2481–2501.

Jung, J., Song, B., Watson, K. and Uslaender, T. (2017), Design of smart factory web services based on the industrial internet of things, *in* 'Proceedings of the 50th Hawaii International Conference on System Sciences', pp. 5941–5946.

Karnouskos, S., Colombo, A. W., Bangemann, T., Manninen, K., Camp, R., Tilly, M., Stluka, P., Jammes, F., Delsing, J. and Eliasson, J. (2012), A soa-based architecture for empowering future collaborative cloud-based industrial automation, *in* 'IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society', pp. 5766–5772.

Kosba, A., Miller, A., Shi, E., Wen, Z. and Papamanthou, C. (2015), 'Hawk: The blockchain model of cryptography and privacy-preserving smart contracts', Cryptology ePrint Archive, Report 2015/675.

Langner, R. (2011), 'Stuxnet: Dissecting a cyberwarfare weapon', *IEEE Security and Privacy* **9**(3), 49–51.

Lee, J., Bagheri, B. and Kao, H.-A. (2015), 'A cyber-physical systems architecture for industry 4.0-based manufacturing systems', *Manufacturing Letters* **3**, 18 – 23.

McKinsey & Company (2015), 'Industry 4.0: How to navigate digitization of the manufacturing sector'.

Merkle, R. C. (1988), A digital signature based on a conventional encryption function, *in* 'A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology', CRYPTO '87, Springer-Verlag, London, UK, UK, pp. 369–378.

Miers, I., Garman, C., Green, M. and Rubin, A. D. (2013), Zerocoin: Anonymous distributed e-cash from bitcoin, *in* 'Security and Privacy (SP), 2013 IEEE Symposium on', pp. 397–411.

Monostori, L., Kdr, B., Bauernhansl, T., Kondoh, S., Kumara, S., Reinhart, G., Sauer, O., Schuh, G., Sihn, W. and Ueda, K. (2016), 'Cyber-physical systems in manufacturing', {*CIRP*} *Annals - Manufacturing Technology* **65**(2), 621 – 641.

Nakamoto, S. (2008), 'Bitcoin: A peer-to-peer electronic cash system'.

Nicholson, A., Webber, S., Dyer, S., Patel, T. and Janicke, H. (2012), 'Scada security in the light of cyber-warfare', *Comput. Secur.* **31**(4), 418–436.

Ning, H., Liu, H. and Yang, L. T. (2013), 'Cyberentity security in the internet of things', *Computer* **46**(4), 46–53.

Scandariato, R., Wuyts, K. and Joosen, W. (2015), 'A descriptive study of microsoft's threat modeling technique', *Requir. Eng.* **20**(2), 163–180.

Schleipen, M., Selyansky, E., Henssen, R. and Bischoff, T. (2015), Multi-level user and role concept for a secure plug-and-work based on opc ua and automationml, *in* '2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)', pp. 1–4.

Schuh, G., Potente, T., Wesch-Potente, C., Weber, A. R. and Prote, J.-P. (2014), 'Collaboration mechanisms to increase productivity in the context of industrie 4.0', *Procedia CIRP* **19**, 51 – 56.

Sicari, S., Rizzardi, A., Grieco, L. and Coen-Porisini, A. (2015), 'Security, privacy and trust in internet of things: The road ahead', *Computer Networks* **76**, 146 – 164.

Swan, M. (2015), *Blockchain: Blueprint for a new economy*, " O'Reilly Media, Inc.".

Tschorsch, F. and Scheuermann, B. (2016), 'Bitcoin and beyond: A technical survey on decentralized digital currencies', *IEEE Communications Surveys Tutorials* **PP**(99), 1–1.

Wang, S., Wan, J., Li, D. and Zhang, C. (2016), 'Implementing smart factory of industrie 4.0: An outlook', *IJDSN* **2016**, 3159805:1–3159805:10.

Wu, K., Li, Y., Chen, L. and Wang, Z. (2015), 'Research of integrity and authentication in opc ua communication using whirlpool hash function', *Applied Sciences* **5**(3), 446–458.

Yang, C. and Wei, H. (2013), 'The effect of supply chain security management on security performance in container shipping operations', *Supply Chain Management: An International Journal* **18**(1), 74–85.

Zhu, B., Joseph, A. and Sastry, S. (2011), A taxonomy of cyber attacks on scada systems, *in* 'Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing', pp. 380–388.

Zyskind, G., Nathan, O. and Pentland, A. . (2015), Decentralizing privacy: Using blockchain to protect personal data, *in* 'Security and Privacy Workshops (SPW), 2015 IEEE', pp. 180–184.