IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 February 2023, accepted 13 March 2023, date of publication 22 March 2023, date of current version 28 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3260621

RESEARCH ARTICLE

SIGNED: Smart cIty diGital twiN vErifiable Data Framework

ZEESHAN PERVEZ[1], (Senior Member, IEEE), ZAHEER KHAN[2], ABDUL GHAFOOR[3], (Member, IEEE), AND KAMRAN SOOMRO[2], (Member, IEEE)

[1]School of Computing, Engineering and Physical Sciences, University of the West of Scotland, PA1 2BE Paisley, U.K.
[2]School of Computing and Creative Technologies, College of Art, Technology and Environment, University of the West of England, BS16 1QY Bristol, U.K.
[3]Department of Industrial Systems, RISE Research Institutes of Sweden AB, 16440 Kista, Sweden

Corresponding author: Zeeshan Pervez (zeeshan.pervez@uws.ac.uk)

ABSTRACT Smart city digital twins can provide useful insights by making effective use of multidisciplinary urban data from diverse sources. Whilst these insights provide new information that helps cities in decision making, verifying the authenticity, integrity, traceability and data ownership across various functional units have become critical characteristics to ensure the data is from an authentic and trustworthy source. However, these characteristics are rarely considered in a digital twin ecosystem. In this research we introduce a novel framework, namely, 'SIGNED: Smart cIty diGital twiN vErifiable Data framework' that is designed on the basis of data ownership, selective disclosure and verifiability principles. Using Verifiable Credentials, SIGNED ensures digital twin data are verifiably authentic i.e., it covers provenance, transparency, and reliability through verifiable presentation. A proof of concept is designed and evaluated based on a smart water management use case to demonstrate the effectiveness of SIGNED in securing verifiable exchange of digital twin data across multiple functional units. The proof-of-concept demonstrates that SIGNED successfully allows the exchange of data in a trusted and verifiable manner at negligible performance cost, thus enhancing security and alleviating privacy issues when sharing data between various functional units in a smart city.

INDEX TERMS Self-verification, web3, blockchain, smart contracts, digital twins.

I. INTRODUCTION

Digital Twin (DT) is an emerging technology that provides a virtual representation of a real object or phenomenon by collecting data from the real object to visualise current behaviour and perform various types of analyses, including predictive analyses to assess future behaviour [1], [2]. DTs are widely being applied in Industry 4.0 [3], manufacturing [4], [5] health [6], medicine [7], construction [8], agriculture and farming [9], [10]. In recent years, the DT phenomenon is gaining momentum in building smart and sustainable future cities [11], [12], [13].

DTs are becoming an important tool for smart cities. Cities present complex systems-of-systems and possess a variety of data such as socio-economic activities, environment, land-use, population, transport, etc. Visualising physical objects into a digital realm with the objective to perform temporal and predictive analyses and derive new information can foster knowledge driven decision making in city governance and policy making. For example, DTs can be used to manage critical resources (water reservoir, power plants), monitor infrastructure (bridges, rail network), operational efficiency (supply chain, traffic management), and meet national / international targets (net-zero, sustainable development goals).

Whilst DTs for cities offer opportunities, Charitonidou [14] examines the critique of digital universalism for urban decision making and highlights that digital universalism is mainly based on theoretical assumptions. A limited set of variables, processes, data, and information on which it is based are necessarily curated. The author highlighted the challenges of data-driven solutions including measurement errors, biases, false positives/negatives, undesired discrimination effects, non-linear dynamics, convergence issues and lack of social dimension. These challenges suggest the

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

need for reliability, trust, transparency, integrity, and privacy embedded in digital twin solutions. Similarly, Raes et al., argue that citizen centric DTs where emotional state or user experience is captured through wearables, especially when interacting with the environment, must consider privacy and security e.g., compliance to General Data Protection Regulations (GDPR) [15].

Deren et al. [13] discussed DT-based smart cities and bidirectional mappings between physical world and virtual world where changes in physical world are automatically reflected in virtual world and vice versa. However, in such a multidimensional environment that will provide an effective way to observe, understand and control the physical world, collaborative data chains pose data trust and security risks. Karaarsalan and Babiker [16] list several security threats and suggest their countermeasures for digital twins including data availability, integrity, confidentiality, unauthorised access, insecure communication, and various security attacks that can undermine the effectiveness of knowledge driven decision making. However, the lack of details on how the proposed countermeasures will be designed in a typical DT architecture requires further investigation. When designing and building DTs for smart cities, it is essential to consider common data security challenges in smart cities. In this respect, Popescul and Radu [17] and Khan Z et al. [18] highlighted end-to-end security including data confidentiality, integrity, identity protection, preserving anonymity e.g., intrusive identification or profiling when data is integrated, consent for use and repurposing, authentication and authorisation, privacy and trust needs. Alshammari et al. [19] also highlighted the lack of security considerations in DTs for smart cities and in particular for Building Information Models (BIM) and made recommendations to incorporate a cybersecurity layer in DTs to deal with various security threats. Khan et al. [20], introduced a blockchain based solution to make use of intrinsic benefits of digital ledger technology to provide verifiable sharing of citizen participation records through permissioned and permission-less ledgers in a smart city environment. Several smart city security challenges and proposed solutions are relevant to design a security framework for a smart city digital twin [17], [18], [20]. Finding solutions to these challenges will ensure the digital assets used in a DT are secure and analyses performed using those assets are traceable, reliable, and trustworthy.

In the above context, this research targets the following security related challenges associated with a smart city digital twin:

1) *Data* ownership and selective disclosure: Since the data can originate from various sources, it is essential that the ownership of data is properly tracked and verified before using it to perform analyses informing smart city applications or services. The data owners should be able to identify data elements for selective disclosure.
2) *Trust, protection and self-verifiable data*: authenticity and trust in the data sources is also essential requirement to ensure data used in a DT model originates from verifiable and trusted data sources. For privacy, the real identity of the data owner is not revealed to maintain the anonymity of the data source. In addition, the data aggregated at various stages of the data production pipeline should maintain data authenticity.
3) *Confidentiality* and integrity: data in transit should be cryptographically protected which ensures that only authorised persons or processes can view and process the data. In addition, the integrity of data is maintained which ensures that the data is not tampered with when it is exchanged between physical and virtual worlds across various stakeholders in a smart city DT.

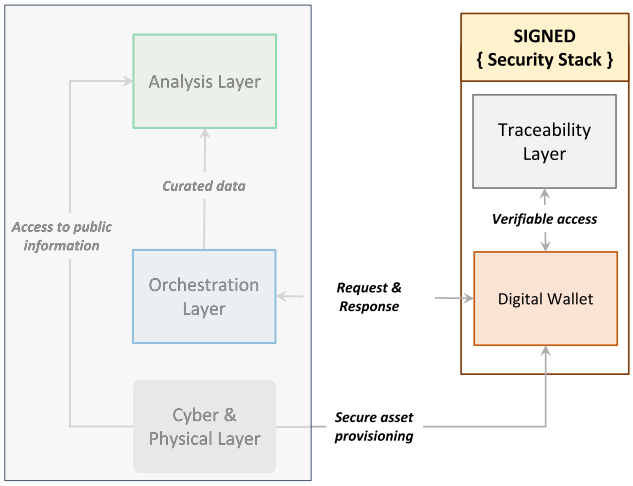

FIGURE 1. High-level conceptual architecture of smart city digital twin, SIGNED providing data verifiability to the digital twin.

With respect to the above security related challenges in smart city DTs, the key research question addressed in this paper is: 'Can a smart city digital twin be designed and built on verifiably protected and authentic data ownership as well as the ability to perform selective data disclosure to manage the data pipeline from diverse data sources?'

To answer the above question and address the security challenges, we introduce a novel framework, namely, '**SIGNED** security stack: **S**mart c**I**ty di**G**ital twi**N** v**E**rifiable **D**ata framework'. The framework is designed on the basis of data ownership and verifiability principles with the aim that digital assets in DTs are verifiably protected. SIGNED not only makes effective use of digital ledger technology but also introduces the concept of self-sovereign identities for verifiable credentials.

Fig. 1 depicts the high-level conceptual architecture of a smart city DT. The Cyber & Physical layer covers both real and virtual objects and data exchange between them. The Analysis layer provides machine learning (ML) and predictive analysis of data. The Orchestration layer communicates with the security stack and is responsible for accessing and sharing verifiable data needed to perform specific analyses. SIGNED is predominantly based on the Security Stack which consists of a Traceability Layer and Digital Wallet. The

Traceability Layer keeps track of digital assets and ensures authorised and verifiable access to those assets. The Digital Wallet provides cryptographic credentials management, creation of verifiable credentials, presentation and their verification for digital assets and DT stakeholders.

A mixed-method approach is used. It starts with performing a rigorous literature review to identify security requirements in digital twin research. To assess the effectiveness of SIGNED security stack, a proof of concept is carefully designed and implemented to demonstrate verifiable data sharing between various functional units independently providing core services to a smart city. The proof of concept is applied on a Water Management use case. The evaluation covers: i) security protocol verification by using scyther tool; and, ii) performance measurement of the solution. These evaluations demonstrate the scalability and effectiveness of SIGNED security stack in secure, verifiable, and traceable sharing of digital assets in a digital twin ecosystem.

The remainder of this paper is as follows: Section II presents related work on security, and use of blockchain with DTs and identifies research gaps. In Section III, SIGNED is presented with detailed architectural design, followed by a description of the overall process in Section IV. Section V presents a proof of concept and in Section VI, SIGNED is evaluated by using a Water Management use case involving a water purification scenario. Several test cases are designed to demonstrate and test the security features. Furthermore, performance evaluation of the SIGNED framework and the results are presented in Section VII. A detailed discussion on SIGNED is followed by conclusions and future work.

II. RELATED WORK

This section captures the state of the art in Digital Twins with emphasis on data sharing, data verification, data ownership, and data security. Alshammari et al. [19] performed a literature review and discussed lack of security considerations in DTs for smart cities, mainly for Building Information Modelling (BIM). They recommended dealing with various security threats by incorporating a cyber security layer in DTs. However, their work lacks details of industrial practices and how to standardise and incorporate security measures in DTs for smart cities.

Shen et al. [21] addressed the secure sharing of big data generated in the lifecycle of equipment, referred to as Big Digital Twin Data (BDTD). The authors highlighted lack of data security and trust amongst stakeholders hampers the benefits of BDTD. To address this issue, the authors proposed blockchain enabled data sharing which utilised cloud storage to persist BDTD encrypted using Proxy Re-Encryption (PRE), and blockchain to store hashes of BDTD. To access BDTD the user generates a blockchain querying request which is returned with the latest address of BDTD stored on the cloud. The proposed work is significantly limited in its functionality and practicality – this is due to the usage of PRE to secure BDTD. For each access request, the data owner has to generate a proxy key to enable transformation of the encrypted data (BDTD) such that it can be deciphered with the private key of the requester. This transformation of BDTD requires the data owner to generate a proxy key for individual users.

Lee et al. [22] addressed the issue of stakeholder fragmentation in the construction sector, which limits their capabilities to share traceable data to effectively analyse 'as-built' and 'as-planned' models in near real-time. They proposed a blockchain based framework which recorded real-time sensor data on the blockchain and projected the sensed data using a visualisation engine. To evaluate the proposed system, the authors utilised Microsoft Azure for blockchain and Internet-of-Things (IoT) integration. Real-time projection of the sensed data was achieved through the Unity Framework. To analyse the progress and compliance of the project, BIM was utilised for comparing the 'as-built' and 'as-planned' models. To secure the shared data digital signatures were utilised. Although blockchain helped to achieve traceability, the proposed methodology predominantly relies on blockchain to secure data sharing with digital signatures. The issues of identity management, revocation, and exploitation are not handled by this work.

Raes et al. [15] presented DUET which proposed a T-Cell framework - a city-wide digital twin framework which facilitates the data exchange of various digital twin models through an Application Programming Interface (API). Individual models are managed as containers, which are accessible through an Apache Kafka instance embedded in the T-Cell architecture. DUET elevates the concept of a digital twin to a city level, hence creating opportunities for data-driven decision-making in tackling challenges faced by city administrations. On-demand integration of digital twins (containers) helps to perform what-if analyses. Although DUET takes a modular approach towards digital twin data sharing/integration, the framework lacks details of digital twin security. For example, it is not clear how an individual digital twin data and services will be managed i.e., ownership, availability, and accessibility; will they be managed centrally, or individual stakeholders can contribute their models – the latter option opens further challenges of provenance and trust amongst others.

Putz et al., [23] addressed the challenges of confidentiality, integrity, and availability when sharing Digital Twin data amongst multiple untrusted parties throughout the assets' lifecycle. A combination of Blockchain (DApp, and DHT), Role-based Access Control, and AES-based encryption was proposed in EtherTwin. It used on-chain and off-chain data sharing to manage the computational load and cost associated with loading, sharing, and processing digital twin data i.e., device specification in the form of AML, and live sensor feed. To demonstrate the practicality, EtherTwin was implemented using Ethereum and Swarm[1] for on-chain and off-chain data storage respectively. Ethereum SmartContracts were used to register digital twins, and manage security keys

[1] Swarm - https://ethersphere.github.io/swarm-home/#

for authentication and authorisation. On-chain and off-chain data management nicely supported the secure sharing of digital twin data; however, this work is limited when managing various phases of the asset lifecycle. For example, when real/physical assets are in scheduled maintenance, access to digital twins should also be adjusted accordingly to the needs of stakeholders – i.e. selective disclosure e.g., the planning department should have access to device specifications, whereas the operations department should have limited access to its live data stream.

Suhail et al., [24] envisioned a three layer architecture for managing the asset lifecycle through the integration of Industrial Internet of Things (IIoT) with blockchain. They proposed, data layer for curating data from physical assets and their virtual representation; storage layer is used to persist curated data on blockchain; and application layer for analysing and developing insight through IIoT. This conceptual framework relies on the intrinsic properties of blockchain to provision a trustworthy ecosystem for data collection, curation, and analysis of digital twin data. The framework lacks details on management and interaction of digital twin across various stakeholders having diverse needs and perspectives of digital twin as it traverses through phases of asset lifecycle. The proposed three layers are also limited in their functionalities – for example, it is not clear how security and privacy will be managed across three layers when stakeholders have varying levels of trust across federated domains.

Kendall [25] defines several DT architecture, data, and security related requirements at a very abstract level. For example, access to data shall be controlled by authorisation component; Data owners shall be able to make data visible and available to authorised users; There shall not be a central authorisation mechanism, but it shall be distributed as required in the ecosystem; Protection of the data from malicious or hostile interference; Allow providers to register and specify the information and services that they intend to offer; Protection in place to protect intellectual property and commercially sensitive data; Access to the data remotely or as a local working copy depending on access permissions; and, support the monitoring of the quality of the data. In the architecture, privacy is achieved through attribute based access control while all data and functions are secure. Furthermore, the authentication and trust is provided through trusted digital id provider and the whole architecture is based on zero trust architecture.

In short, the above literature review indicates several research gaps related to security and privacy in DTs. Most of the focus appears to be on securing the exchange of data, but very little thought appears to have been afforded to the issues of identity management, trust and authorisation. Therefore, we propose a solution to these issues in the following section.

III. SIGNED SECURITY STACK ARCHITECTURE AND DESIGN

To put the SIGNED security stack into perspective, the proposed Smart City Digital Twin architecture is designed into five cohesive functional components; namely: Cyber & Physical Layer, Workflow Designer, Analysis Layer, Traceability Layer, and Digital Wallet. The combination of Traceability Layer and Digital Wallet form a SIGNED (security stack) which provides the functionality of safeguarding assets and ensuring analysis drawn can be traced back to original data sources, and data owners have control over their shared data i.e., verifiable, trustworthy, and secure. Fig. 2 presents the component level details of the Smart City Digital Twin, and illustrates the core role of SIGNED security stack in achieving verifiable data sharing across various actors.

The main focus of this paper will be on the SIGNED security stack, and it will be covered in detail in later sections. Cyber & Physical Layer, Orchestration Layer, and Analysis Layers are briefly presented to provide full context to the SIGNED security stack but these layers are beyond the scope of this paper.

1) ***Digital Wallet:*** Each stakeholder or functional unit in the digital twin data pipeline requires a wallet which provides cryptographic credentials management, creation of verifiable credentials, presentation and their verification. The digital wallet provides the core functionality of security through verifiable credentials and presentation, those are based on Self sovereign identity concepts [26]. It ensures that each access to assets conforms to security, integrity, trust and data ownership requirements. This helps to curate data from authentic digital and physical resources, thus analysis drawn from it is considered trustworthy and traceable. The Digital Wallet provides the following core functionalities:

   a) *Cryptographic Credentials Management:* The core function of the wallet is to create, manage and register cryptographic credentials. Creation means creating a Blockchain account; management refers to securely storing and making available the private credentials (key pair) for encryption and verification; and, registration means to register public credentials along with a decentralised identifier in the Verifiable Data Registry (VDR) i.e., a smart contract-based registry in blockchain. Public cryptographic credentials should be available to all through decentralised oracles.

   b) *Verifiable Object Creator:* By using the cryptographic credentials, the Digital Wallet helps to create an anonymous verifiable object which will be shared between the entities (or actors) in the data pipeline. In this way, these credentials will be used to establish an anonymous secure tunnel between the data sharing parties without revealing their identity.

   c) *Verifiable Object Verifier:* This component implements the logic to verify the received verifiable credentials.

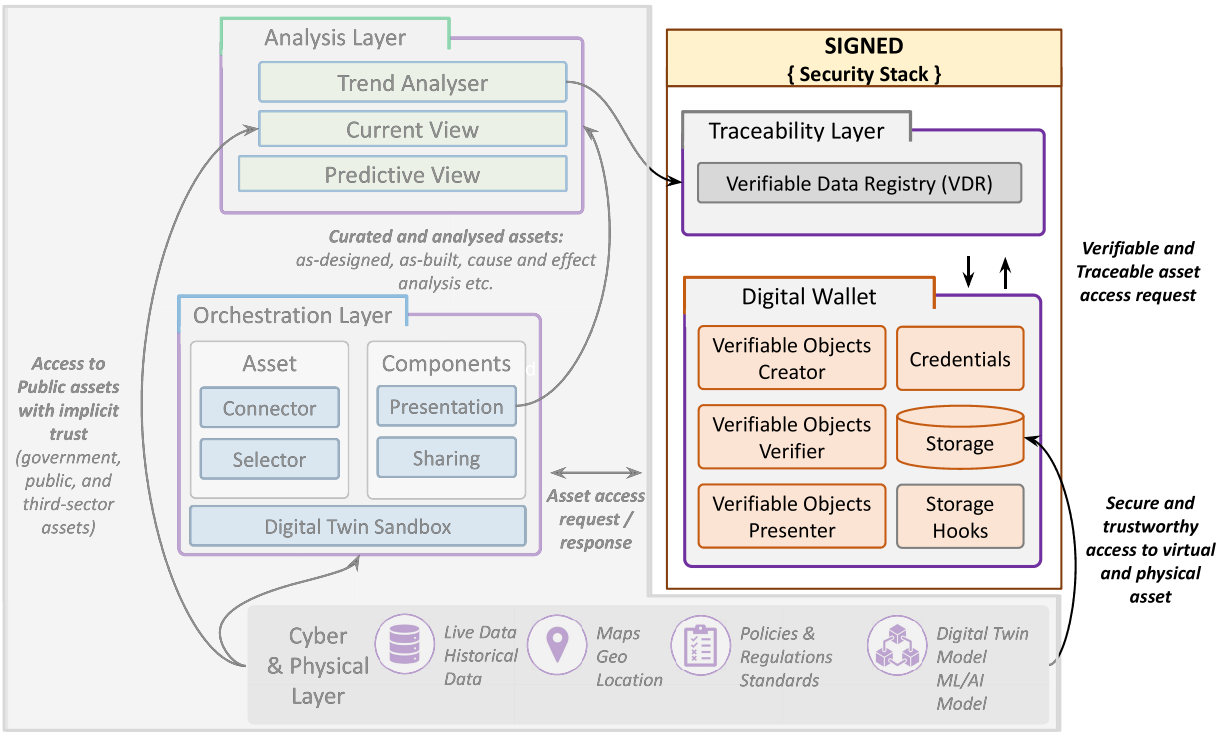

FIGURE 2. Architecture of SIGNED security stack along with its deployment in Smart City Digital Twin for verifiable and secure data exchange within. The primary focus of this paper is on verifiability and security of digital assets used in DT, through SIGNED security stack.

d) *Verifiable Objects Presenter:* This component implements the algorithm for selective disclosure by selecting, sharing and presenting a subset of attributes (data that need to be shared or requested) to the requester (in this case, verifier or data consumer). This component extracts the subset of attributes from the verifiable credentials and then creates a verifiable presentation which comprises attributes and proofs to implement selective disclosure. In the final verifiable presentation, one proof can be directly copied from the original verifiable credentials which shows the origin of the data while the other proof shows that the holder of the verifiable credential is presenting the requested data. This unique approach helps in building a self verification and trusted data sharing infrastructure for the SIGNED security stack.

2) ***Traceability Layer:*** It keeps track of registered digital and physical assets in a Smart City Digital Twin. It holds the Verifiable Data Registry (VDR) component which is a smart contract deployed on the blockchain and is responsible for managing and sharing the public credentials of the components such as public keys and public addresses. These credentials are used for creating verifiable data credentials and public keys are part of the verifiable data credentials for verification purposes. In addition, technically those public keys must be in the verifiable presentation to achieve traceability of the verifiable data credentials, for example, information about who created verifiable credentials, and who is the presenter of the credentials. Of course, It ensures that security and privacy requirements are enforced such as preventing exploitation of the digital assets and stakeholders, thus enforcing intrinsic traceability property of blockchain.

3) **Cyber & Physical Layer:** This layer contains the information about cyber and physical assets registered and accessible through Traceability Layer. It also serves as a repository to register assets, which can then be curated for analytical purposes in knowledge driven decision making.

4) **Orchestration Layer:** This layer helps users to plan interactions amongst various assets (e.g., data and Machine Learning models) and allow them to try various configurations of assets from Cyber and Physical Layer in order to achieve the desirable output (analysis). It has the following components:

a) *Digital Twin Sandbox:* It provides a safe execution environment for testing various workflows and making necessary enhancements and optimisation based on user needs.

b) *Assets:* This component comprises selectors and connectors. A selector helps users to identify assets which are compliant to the needs of users. The connectors provide necessary information

and means to access the assets from a Digital Market Place.

c) *Components:* handles presentation and sharing of digital twin assets and outcomes. Presentation facilitates the users in demonstrating the workflow in various formats aligned with the needs of the target audience. Sharing enables users to disseminate their workflow with relevant peers, groups, and venues.

5) **Analysis Layer:** This layer provides insight analysis drawn from the individual assets or their combinations. It provides templates for various analysis types such as, as-designed analysis to simulate assets in a given environment (context), as-built to perform live monitoring of asset, cause and effect analysis to examine assets' performance in a changed environment, predictive analysis, and trend analysis etc.

IV. SIGNED SECURITY STACK DESIGN SPECIFICATION

This section presents the design and algorithmic details of SIGNED security stack.

The overall process is shown in Fig. 3:

1) It starts with the deployment of a Verifiable Data Registry (VDR) to manage public credentials, which will be used for sharing Verifiable Credentials (VCs) between different functional units.
2) Key Agreement VCs are created between sender and receiver to establish a secure session so that they can securely share data.
3) Issuance of Data VCs follows the Key Agreement VC and is used to securely exchange data between two functional units.
4) The VCs themselves are also verified because they are digitally signed.
5) A Verifiable Presentation (VP) is created and sent to the requester.

Three roles are defined to keep the SIGNED security protocol simple. The Issuer is the owner of the data who creates secure credentials with the Holder as well as shares the data. The Holder is a central authority which can be used for verification and data sharing purposes. It is either owner of the data or custodian of the data. The Requester/Recipient is a functional unit which requests and receives specific data through Holder.

A. DEPLOYMENT OF VERIFIABLE DATA REGISTRY (VDR)

The VDR manages public credentials of users or functional units, and we assume for the purposes of Proof of Concept (PoC) that the VDR is already deployed. The VDR provides the key-value based structure shown in Listing 1 to manage RSA public keys of each functional unit or user.

This structure is stored in the VDR against the public address (e.g., Ethereum address) of each user. In this system, the key is formatted as *did:veid:enterun-address* where *veid* refers to veidblock [27]. The key-value based structure stores

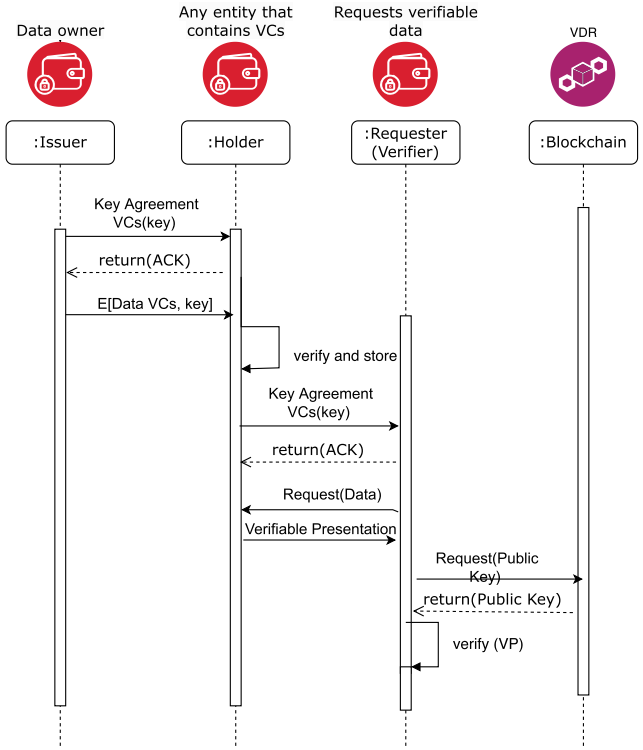

FIGURE 3. High-level process. It assumes that the VDR is already deployed.

Listing 1 Key-Value structure for VDR credentials

```
[
  {
    "address":
↪   "did:veid:0xC59b.....C3C",
    "rsaPublicKey":
↪   "MIIBIjAN.....TIa4LwIDAQAB"
  },
  {
    "address": "did:veid:0x71.....8039",

    "rsaPublicKey":
↪   "MIIBIj.....kZvKlQIDAQAB"
  }
]
```

the *key id (blockchain address)* and *public key* generated by the functional unit. As the system operates in an open and decentralised environment, this means that any functional unit in a smart city can join, create, and register RSA public key in the VDR. Furthermore, if anyone wants to communicate or issue verifiable credentials, then they must have the public address of the recipients.

B. ESTABLISHING SECURE SESSION USING KeyAgreement VERIFIABLE CREDENTIALS (KA VC)

The KA VC process takes place between a sender (e.g., Issuer, Holder) and receiver (e.g., Holder, Requester). The

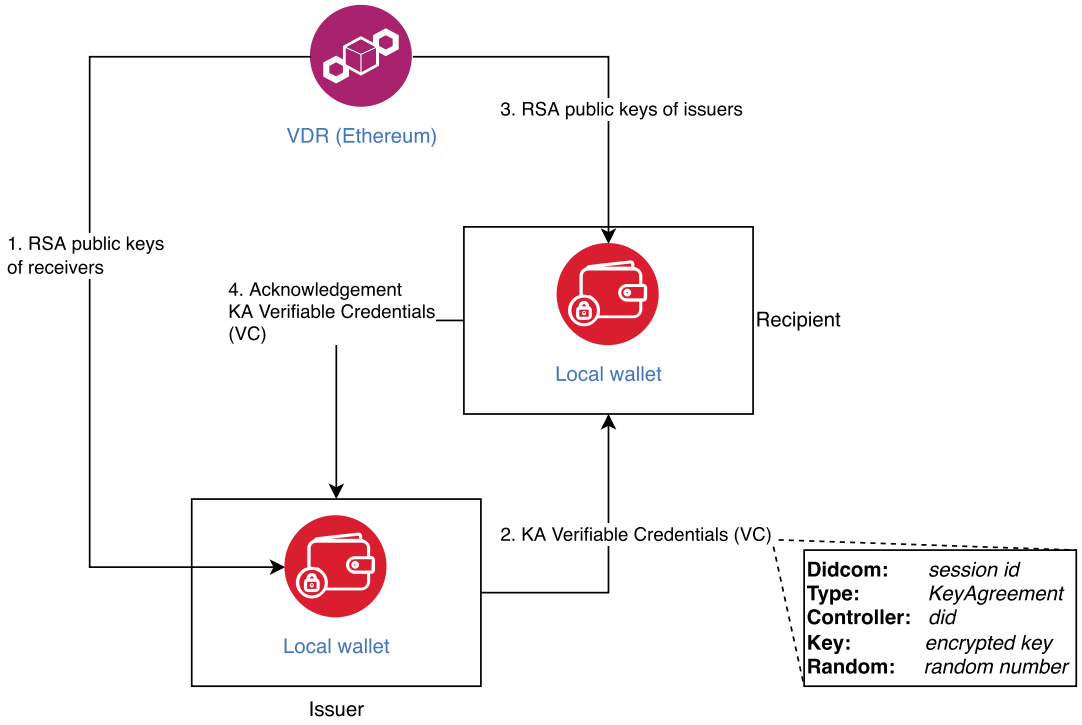

FIGURE 4. Establishing secure session using KA VCs.

Listing 2 Formal KA VC Process; where e = encrypted symmetric key encoded in BASE64, p = digital signature, puk = public key, prk = private key, nonce = single-use random number, ec = encrypted and digitally signed message, E = Encryption function, H = Hash function, skey = symmetric key, '|' = pipe used as delimiter, D=Decryption function

```
e` = BASE64e(E(skey, puk))
p = E(H(e`|nonce), prk)
ec = e`| nonce |didcom(H(e`))
skey = SubString(D(BASE64d (e`),prk),
 ↪  '|')
```

overall KA VC process is shown in Fig. 4. The PoC uses *web3* technology where every organisation (i.e., functional unit) downloads the wallet to communicate with the decentralised network and has direct communication with its peers to share verifiable data credentials. In the initial setup, the wallet must be configured so it creates a VDR account (in this PoC, it is an Ethereum account) and then it registers RSA public keys in the VDR. When an organisation shares data with another organisation, it starts the process by establishing a *didcom*. As both organisations have yet not established trust and a secure channel, therefore, first they must authenticate and exchange a shared secret to establish a secure session. For this purpose, they use a request-response protocol. In the request, the sender creates a *'KeyAgreement'* VC in which its *claim* has the following attributes:

- *Didcom*: It is a unique decentralised identifier to identify session id
- *Type*: Type of claim (e.g., KeyAgreement)
- *Controller*: This is a unique *did* to identify the issuer and owner of the credentials
- *Key*: Encrypted shared symmetric key encoded in BASE64 format
- *Random*: A random number to avoid a replay attack.

In the KA VC process in Fig. 4, two attributes are important to establish a secure session, *Key* and *Random number*. The sender generates a symmetric key and then fetches the RSA public key of the recipients from the VDR. It encrypts the *Key* by using the receiver's public key and then encodes it into BASE64 as shown in Listing 2. The *Random number* is randomly generated, which resists the protocol against replay attack. The sender also includes the verification method, calculates proof and attaches it with the claim along with other standard attributes such as issuer address, controller, type of verifiable credentials, etc.

The sender sends this KA VC to the recipient. The recipient receives it and extracts the issuer address. After that, it consults the VDR and fetches the public key stored against the issuer address. By using this public key, the receiver verifies the KA VC and then extracts the claim part from which it extracts the *didcom*, *Key*, and *Random number*. It decodes the key and then decrypts using its own private key. After that it saves the shared key in the local *wallet* along with *didcom* while it creates another KA VC which will be used as an acknowledgement. Here we must note that the random

number is encrypted with the shared secret, so the sender can also mutually authenticate and protect the channel from replay and impersonation attacks.

C. SECURE ISSUANCE OF DATA VCs

Now the shared secret is exchanged, the issuer creates VCs based on the mDL standard [28] by following the steps presented in the algorithm Algorithm 1.

Algorithm 1 Creating a Data VC

```
1: procedure CreateDataVC(issuer, holder, claims, keypair)
2:    i=0, digestContainer[]
3:    for all claims do
4:       salt = generatesalt()
5:       saltedClaims[i] = attributeobject(claims[i].key, claims[i].value, salt)
6:       dc = digest(saltedClaims[i])
7:       digestContainer[i] = dc
8:    end for
9:    json = formatVC(issuer, holder, digestContainer, "VerifiableCredential_Header", keypair.publickey)
10:   signedJson = proof(json, keypair.privatekey)
11:   encrypedClaims = encrypt(saltedClaims, shared-key)
12:   vc = signedJson + encrypedClaims
13:   return vc
14: end procedure
```

According to Algorithm 1, following is the complete process to creates a verifiable credential both KA VC and Data Verifiable Credentials:

1) The algorithm takes an issuer, holder, key-pair, and claims as inputs to create VC. In all four attributes, claim is the most important one, which is a set of attributes needed to be shared and digitally signed. First, it calculates digest of each attribute of claim to fill digest container. In the mDL standard, a digest is created for every attribute and uses a salt, attribute name and value as an input to the digest function. This produces an attribute object shown in Fig. 5. In the next step, a unique sequence number is assigned to each attribute, as depicted in Fig. 5 Claim *"O1"*. For each attribute, step 3 to step 6 are repeated. After that a claim object is created based on the following steps.
2) Insert each digest created in Step 6 into the digest container as shown in step 7 (a key-value structure) where the key is the attribute object number, and the value is the digest of the object.
3) Add digest container in the VC (Step 9)
4) Calculate proof or digital signature in which it considers all VC attributes mentioned in step 1 and digest container. These are all formatted in JSON object. After that, it inserts the proof object into the VC.
5) In order to securely exchange the VC over an unprotected channel, the issuer encrypts the saltedclaims (claim object) by using the shared secret obtained during the process of establishing the *didcom*.
6) Creates encryptedClaims which contain the following attributes in JSON format. It attaches encrypted claims with the VC to form a complete digitally signed verifiable credentials.

```
"claims": {
    "encryptedClaims":
    "SUU5dDE3cjRzRGRwOT0=",
    "didcom":
    "did:veid:0x767389a9889b87cb8a",
    "encoding": "BASE64",
    "type":
    "AES128/AES/CBC/PKCS5Padding"
}
```

In the next step, the issuer sends the created verifiable credentials to the holder (i.e. a broker which takes the role of data custodian).

D. VERIFICATION OF VERIFIABLE CREDENTIALS

The holder receives the VC and then performs the following steps to verify and open/review VC:

1) The holder extracts the issuer's *did* from VC.
2) It fetches the public key of the issuer *did* from the VDR.
3) It also extracts the public key from the poof object of the VC.
4) It compares both public keys, if they are the same keys then the holder de-attaches the claim and then verifies the proof.
5) If the proof is verified, then the holder processes the *encryptedClaim* to extract the claim object. It extracts the *didcom* from the claim object and then fetches the shared secret from local protected storage (e.g., wallet). Then it decrypts the encrypted claim and verifies the individual attribute digest with the corresponding digest values given in the digest container.
6) In the next step, the holder attaches the claims with the VC before storing it in the local wallet.

All the objects between the holder and the object issuers follow the same process to exchange secure and verifiable data objects.

E. VERIFIABLE PRESENTATION FOR THE REQUESTER

We adopted the *selective disclosure* approach, where only the requested attributes of the Issuer's data can be shared with the Requester. The holder fetches the VC along with claims and then copies the VC header, proof and digest container in a new object and selects the subset of attributes (only those holders is willing to share) from the claim and creates a *Presentation Object*. The presentation object is digitally signed by the

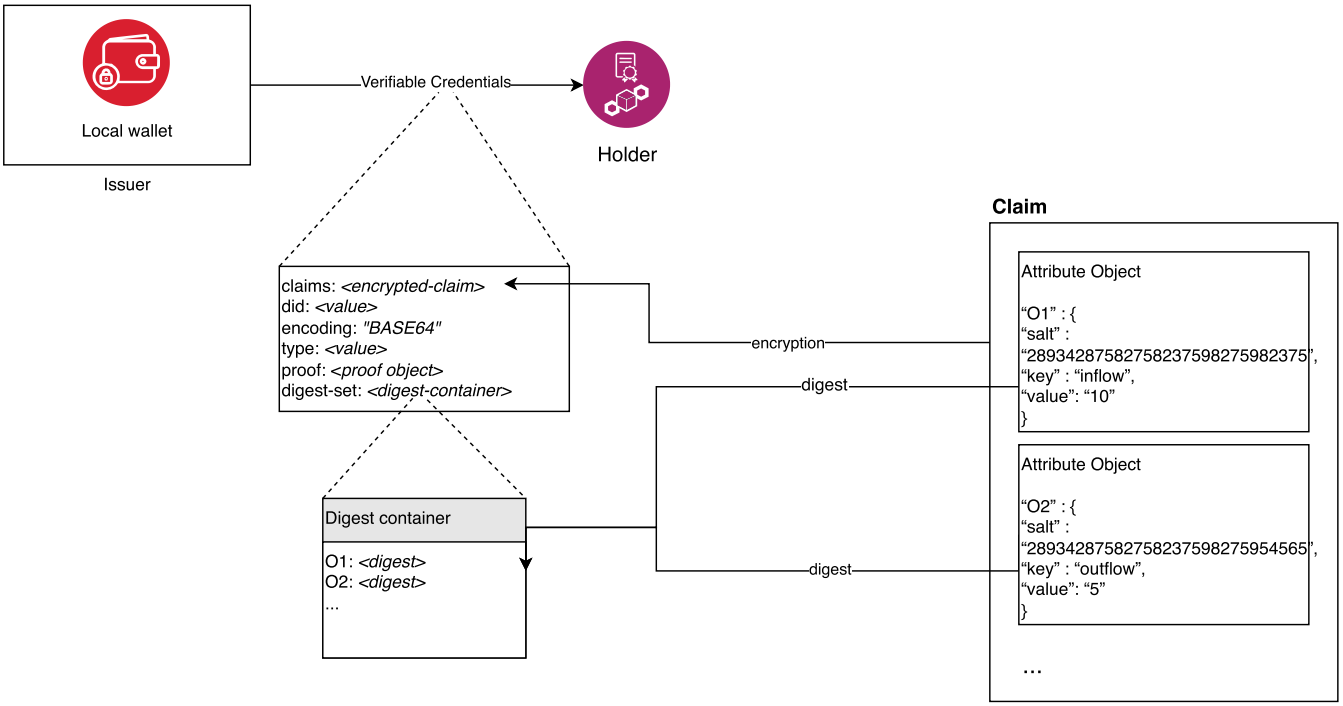

FIGURE 5. Securely issuing a data VC.

holder. The holder now encrypts the claim by following the same steps used by the issuer in the previous section. After that, it sends it to the Requester.

Now the Requester can verify the presentation and then also is able to verify the sources of the original VC. In this case, it trusts the received data because its source is verified, its presenter is verified, data integrity and confidentiality are also ensured. In addition, the verifiable credentials also contain information about the issuer, holder, and requester which helps to provide tractability of the credentials. We also mentioned here that the identities of the various actors are not exposed since they are in the form of *dids*.

V. PROOF OF CONCEPT THROUGH SMART WATER MANAGEMENT USE CASE

To facilitate a better understanding of the PoC in realising verifiable and secure data sharing in DTs, a Smart Water Management use case is used and illustrated in Fig. 6 and an overview of the flow of activities is shown in Fig. Fig. 7.

River Parrburn is the main source of water for the inhabitants of Aysgarth City. The river serves the domestic water needs, irrigates the surrounding agricultural land, and generates power through the run of the river. Considering the strategic importance of the river, **Aysgarth City Council** (ACC) has decided to continuously monitor the water flow in River Parrburn, and cohesively manage its independent functional units through a Digital Twin ecosystem. These functional units are '**river flow management**' for monitoring of water which flows in and out of the river, '**water purification**' for domestic water usage, and '**water stream management**' for run of the river **power generation**. These functional units perform their operations through various sensors and actuators, managed by independent agencies (or organisations). Due to inherent independent management of these functional units, data (sensors, policies, and procedures) resides in disjoint silos, resulting in various security, ownership, privacy and trust issues when sharing the data with other agencies. To create a Digital Twin of River Parrburn, ACC addresses these issues with the SIGNED security stack. The Digital Wallet of the SIGNED security stack enables all agencies to push their self-verifiable data to the Digital Twin securely whilst independently controlling the data exposure i.e., which data to share, how frequently to share and, how to present to others, etc.

For demonstration, we defined the following high level data model (Fig. 8 with core attributes).

For better readability, Table 1 presents the mapping and synergy between terminologies used in the SIGNED security stack and standard security protocol.

A. PoC TESTING SCENARIO

For proof of concept, we defined four main roles: WaterStreamMgnt (Issuer), ACC (Holder), Citizen and WaterPurification Company (Requester/Recipient) for the following test scenario derived from the above water management use case:

*The **Water Purification Company** relies on the import of chemicals used in various procedures of water treatment such as: Coagulation and Flocculation etc. Maintaining a decent*

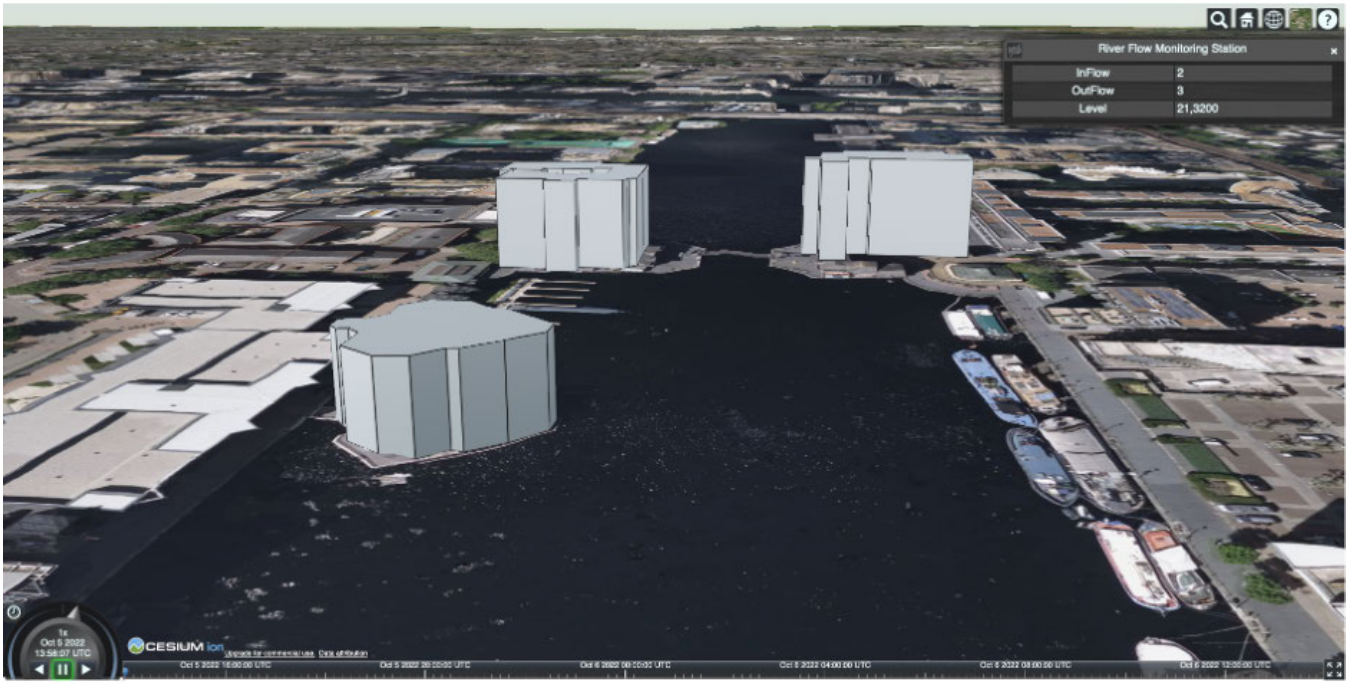

FIGURE 6. Digital twin showing monitoring stations reporting various parameters (Figure produced using Cesium ion).

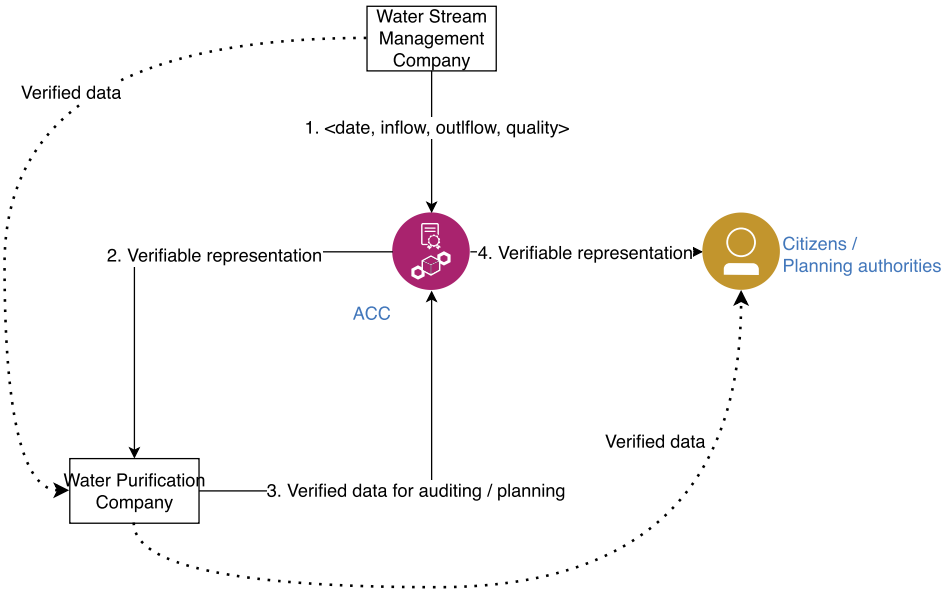

FIGURE 7. Various entities sharing and querying verified data.

stock of those chemicals is becoming a critical issue for the Water Purification Company, especially, in the midst of higher fuel prices and supply chain uncertainties. To ensure frictionless operations of the water treatment plant, the Water Purification Company requests the River Flow Data along with Sensor Information measuring water quality from the ***ACC*** *which holds the VCs. The ACC will create a verifiable presentation which includes the date, inflow, outflow and quality of water attributes from the original VCs. The Water Purification Company can verify the verifiable presentation and also verify that the data is generated by the* ***Water Stream Management Company*** *which will develop trust in the data originality. In this way, trust, authenticity and verifiability of the data will be provided to the Water Purification Company. In addition, since the ACC is involved in water planning in the city therefore it needs updated information from the*

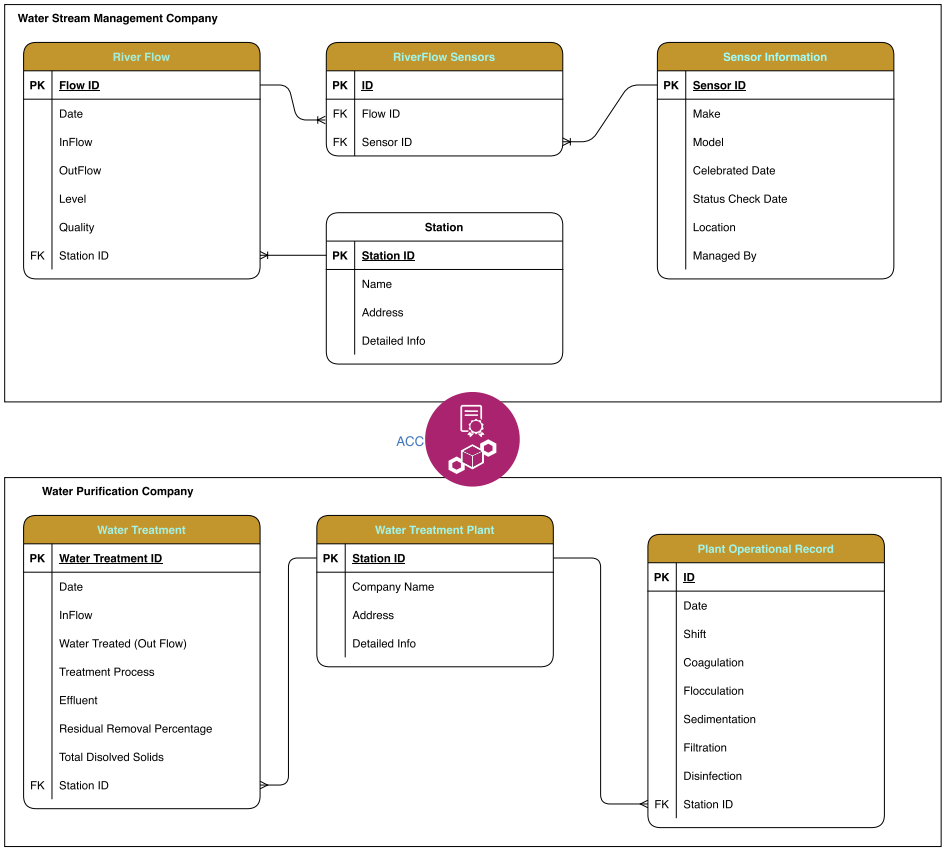

FIGURE 8. Use case data model.

*Water Purification Company. Furthermore, if a **citizen** needs to know about how the water is purified then the ACC is obliged to provide them authentic information. For all such needs, the Water Purification Company periodically creates a VC and then sends it to the ACC where ACC uses these VCs to create verifiable presentations for water planning and citizens.*

B. PROOF OF CONCEPT DEMONSTRATION

Technologies: The proof of concept of defined usecase is implemented by using Java [29], Spring Boot [30] and web3j libraries [31]. Furthermore, we used ganache for VDR (i.e., smart contract deployment) [32] because it uses less resources as compared to Ethereum blockchain [33] private network. We deployed our developed solution on Ubuntu 20.04 LTS which has 8GB memory and Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz processor.[2]

Deployment: First, we deployed ganache and generated a smart contract (VDR) in ABI format using Truffle [34], [35]. We also generated Java objects for the VDR using web3j auto generated converter. After that we created a local account and deployed VDR on blockchain which took 1216 ms. Furthermore, we created accounts for acc, river flow management company, water purification company, and one citizen. We registered their RSA public keys in the VDR which took 211 ms on average, while fetching same credentials took 65 ms as shown in Table 2. RSA public keys registration is a one time activity, but fetching the RSA public key can be performed multiple times, and it is negligible.

In addition to the deployment of supporting component modules, we also developed following applications. Further detail of these applications is given in the section Section VI-B.

- River flow management company
- ACC
- Water purification company
- Citizen

VI. EVALUATION RESULTS AND DISCUSSION

Key agreement, verifiable credential exchange, and selective disclosure are the core functionalities to realise verifiable and secure data management in a digital twin. These functionalities also serve as the building blocks to test verifiable and secure data management in a smart city DT ecosystem. The proof of concept implementation of the SIGNED framework and its application on the water management use case covered in the previous section demonstrates these functionalities.

[2]For provenance, the KeyAgreement VC and the Data VC source code is available at the github link:
https://github.com/aghafoor77/digitlatwin/tree/main

TABLE 1. Mapping and synergy between standard and SIGNED terminologies.

Standard Security Terminology	SIGNED Security Stack Terminologies
Organisation	Functional Unit in Smart City Digital Twin
Recipients	Organisations (or individuals) which request and receive selective data owned by other organisations (or individuals) or through a data custodian (e.g, Holder)
RSA Key Pair	Public-private key pair
Key Exchange	Sharing of symmetric key between Issuer and Receiver/Requester
Wallet	Digital Wallet from SIGNED security stack
Claim in VC context	Claim refers to the atomic data items owned by a functional unit that may need to be shared with another functional unit
Scyther Claim	These are the functional properties of the security protocol. Each claim asserts that a specific SIGNED functionality is secure against specified security attack(s)
Proof	Proof contains digital signatures to verify a data object
Presentation Object	Verifiable Presentation Object
Blockchain/Distributed Ledger e.g., Ethereum	Verifiable Data Registry (VDR) is a data structure deployed on Blockchain
Issuer/Sender or Receiver/Requester, i.e. an organisation or actor who can send or request data	Functional unit
Holder	The Holder is a central authority which can be used for verification and data sharing purposes.
Security Protocol	SIGNED security stack functionalities which help in achieving secure key agreement, ownership, verifiable credentials and selective data disclosure

TABLE 2. VDR Deployment and Public Key Handling Performance.

Activity	Time (ms)
VDR Deployment	1216
RSA Public Key Registration	211
Fetching RSA Public Key	65

Now in this section, those core functionalities are choreographed as a secure data sharing protocol and empirically evaluated by using a verification tool Scyther [36]. Furthermore, performance evaluation is carried out by using the proof of concept implementation and up-scaling the number of functional units.

A. SECURITY EVALUATION

Scyther is the most popular tool used for formal verification of security protocols. To demonstrate the effectiveness of the SIGNED framework against common attacks, Scyther's built-in adversary model is applied. The adversary model launches attacks in the sequence of protocols and then verifies its scyther claims. The scyther based SIGNED framework evaluation helped to verify the strengths and weaknesses of the protocol.

To evaluate the verifiable and secure data management, the following scyther claims are applied. These scyther claims evaluate the internal working and resilience of the overall protocol against impersonation, and replay attacks. Furthermore, the protocol was also evaluated for the confidentiality and integrity of the exchanged data:

- *Secret*: we defined Secret claim which fulfils the requirements to securely exchange symmetric key;
- *Nisynch*, ensures the communication between sender and receiver is synched and sent by the sender;
- *Alive*, is required for authentication and to ensure the recipient received messages and processed it;
- *Commit*, is used for commitments between sender and receiver and makes effective claim against ***impersonation attack***. We also compare the nonce exchanged between the WaterStreamMgnt and ACC by using match function i.e. match(n,n1);
- *Niagree*, ensures that the non-injective property is achieved to protect protocol from ***replay attack***.

As shown in Fig. 9 all scyther claims are verified by scyther verification process.[3]

WaterStreamMgnt and ACC first executed the key agreement protocol, which must ensure the secrecy of the symmetric key and provide protection against replay attacks. Then the same protocol was extended, and verifiable credentials were created. These verifiable credentials included *inFlow*, *outFlow*, *quality* attributes and were digitally signed by using the secret key (private key of WaterStreamMgnt) while encrypting using the shared secret. In this exchange, *Nisynch*, *Alive* and *Niagree* were claimed. These provided sync communication, the recipient received verifiable data credentials and ensured that the sender is valid. The encryption of the attributes ensured that the data verifiable credentials are protected against the disclosure of the value of attributes.

Now the WaterPurificationComp requested for *inFlow* information while the Citizen asked for the *quality* of the water from the ACC. The ACC created two verifiable presentations. One for WaterPurificationComp and second for Citizen, which contains only the required parameters. In both cases, *Nisynch*, *Alive* and *Niagree* were claimed. As depicted in Fig. 9, the execution of the protocol verifies that all the claims are fulfilled.

In order to ensure the correctness of the protocol, we intentionally induced an error in the sequence of the protocol. We exchanged *quality* attribute of the water flow which is not shareable and consequently, ACC sends the quality attribute

[3]For provenance, the SIGNED protocol verification scyther code is available at: https://github.com/aghafoor77/digitlatwin/blob/main/DTDemoIHVV2.spdl

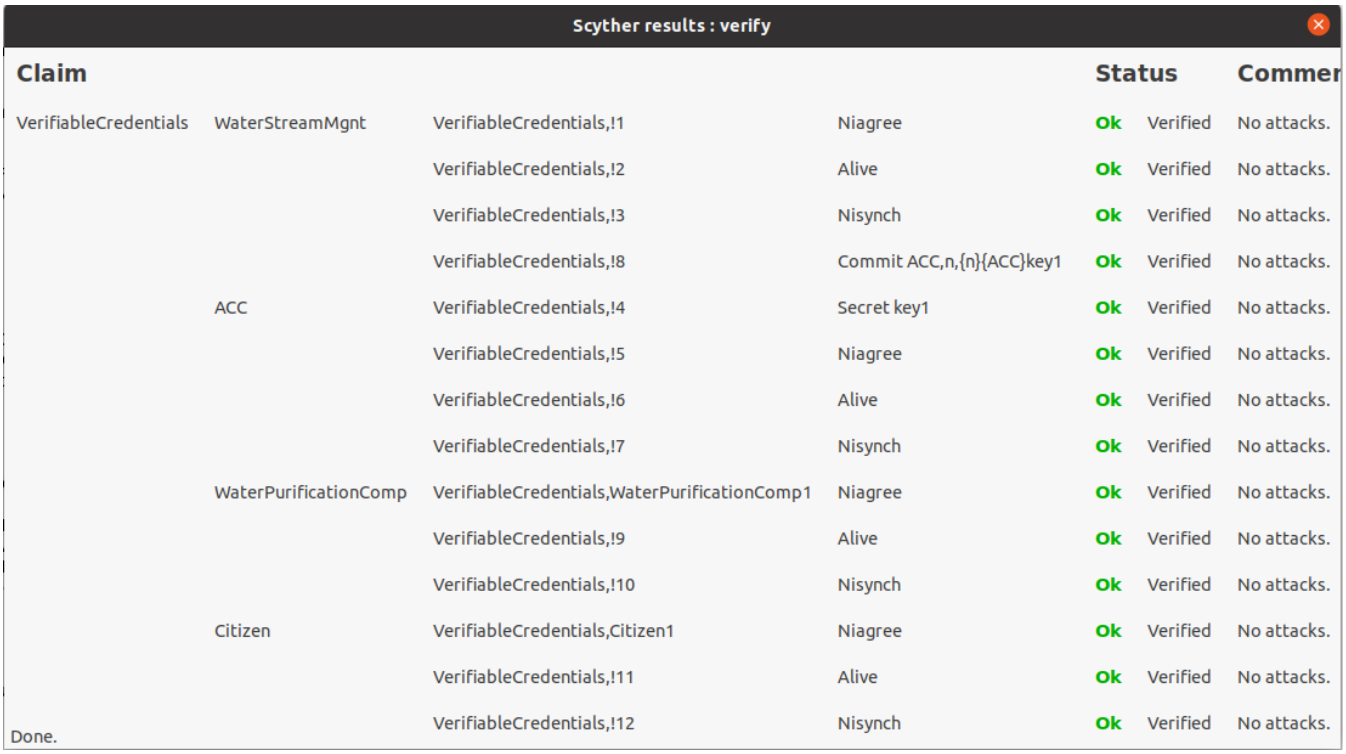

FIGURE 9. Protocol verification of test scenario for selective disclosure between water stream management, ACC, water purification company and citizen by using scyther.

in clear text form (i.e. it is not encrypted) to the Citizen. We analysed that scyther detected the error (i.e. the quality attribute is not encrypted) and failed the required claims as depicted in Fig. 10. It suggests that a man-in-the-middle attack is possible, e.g., an intruder can either view the *quality* attribute or can inject the false value of *quality* attribute in the message which consequently can result in exchange for incorrect data to the Citizen. Hence, it verifies the correctness of the protocol designed for Fig. 9.

Based on the above verification results, the following discussion highlights the SIGNED security properties for secure data management in a smart city DT.

1) DATA OWNERSHIP

Each object (exchanged data: inFlow, outFlow, quality etc.) received by the ACC is cryptographically digitally signed by the object creator, so the object remains the property of the data owner. If any third party is interested to use that object, then the party must request ACC. ACC's consent (being the data custodian) is required to share it with the requester because ACC will create a verifiable presentation of the already verified object (verifiable credentials). This certifiable presentation requires ACC's private key to digitally sign the presentation.

2) VERIFIABILITY

The SIGNED framework requires that all objects (either in the form of credentials or presentation) must be digitally signed so the recipients and/or verifier can verify them. In addition, the identity of the owner can also be verified from the verifiable registry (VDR) which increases the trust of the verifier in the ACC. This is mainly due to public key sharing through the immutable VDR.

3) AUTHENTICITY

Since the verifiable credential and presentation are digitally signed by the owner of the credential and ACC for the presentation, therefore it provides authenticity.

4) DATA DISCLOSURE

All the data credentials and presentation are encrypted using symmetric key cryptography, so the contents of both objects are not disclosed to unauthorized verifiers (functional units).

5) IMPERSONATION

As mentioned in the proof of concept, each functional unit or entity registers its account in the verifiable data registry (VDR) along with the public credentials and the same credentials are used for verification, so there is a very low probability of impersonation since the owner is responsible to register its own credentials.

6) REPLAY ATTACK

During the key agreement, *nonce* is exchanged, which provides resistance against the replay attack. Since the nonce is only valid for live communication and can only be used once, this helps to cater for the replay attack.

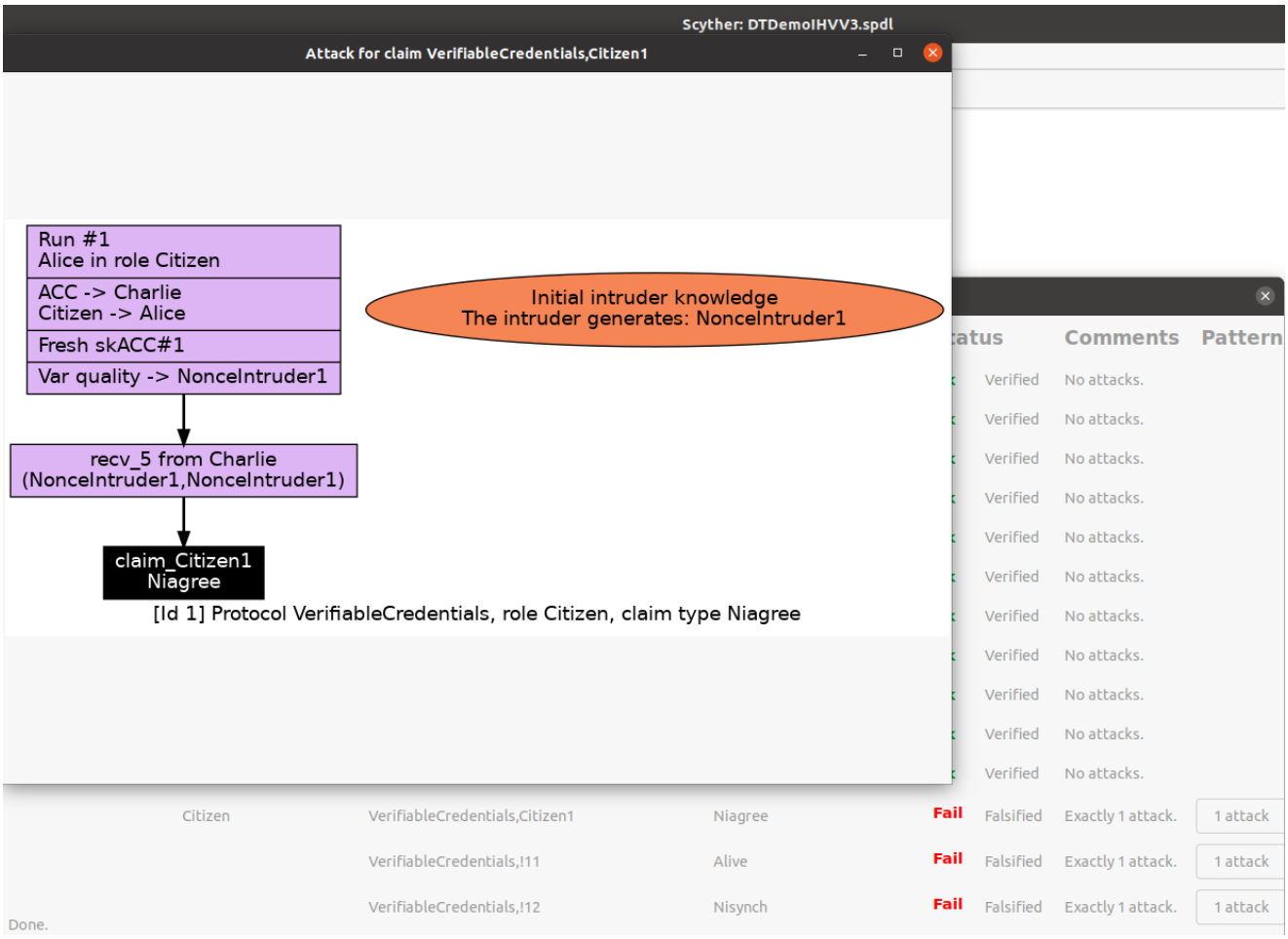

FIGURE 10. Protocol verification of test scenario with attack injection between ACC and citizen by using scyther.

7) PRIVACY

All the data is owned and controlled by the owner. However, requesters can only request data from ACC. This helps to achieve the privacy of the Issuer (data owner) as requests are always encrypted using symmetric encryption. In addition, the proposed system does not use real identities, so the identities of the actual owners are not exposed.

8) TRACEABILITY

In the SIGNED security stack, all the credentials are signed by the Issuer (i.e., original owner) which provides information about the origin of the information. If it is presented to a third party, then it also contains the signature of the presenter (i.e., ACC) and the original owner (i.e., Issuer) which helps the requester to verify the owner and the presenter. Therefore, the SIGNED framework solution provides a traceability feature.

9) IMMUTABILITY

All the public credentials are stored in the VDR and Verifiable Credentials are digitally signed, so it resists content tampering, therefore, the proposed system provides immutability.

The selective disclosure property of the SIGNED framework is also different from other access control solutions. Access control list, role-based access control and attribute-based access control are applied by the centralized authority. They are enforced by the centralized service provider while selective disclosure is based on the concepts of data ownership and the controller of the data has the right to select attributes which he or she wants to share. In addition, it provides data verifiability and authentication as well, so these are the inherent properties of the verifiable credentials when it is presented to the recipients. Most of the existing access control solutions work in the restricted domain while the selective disclosure is based on decentralized VDR, so these can be shared by the owner of data verifiable credentials with any other entity or functional unit for processing.

B. PERFORMANCE EVALUATION

Applications: For demonstration, the following four wallet applications are implemented:

- **River flow management company**: This module is implemented in Java and web3j. It creates a secure

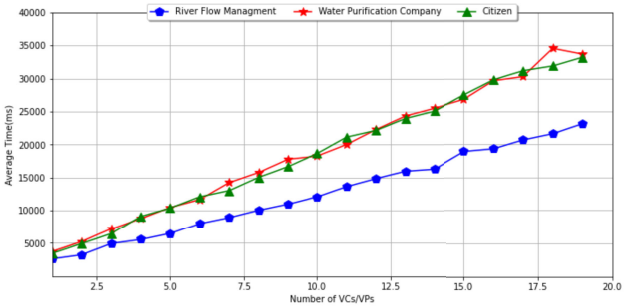

FIGURE 11. Performance analysis of uploading data verifiable credentials and downloading data verifiable data presentation.

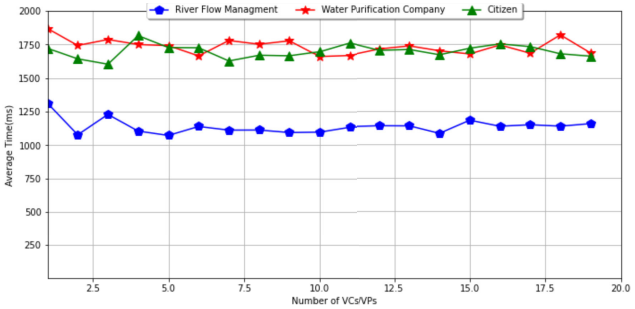

FIGURE 12. Average time comparison of uploading data verifiable credentials and downloading data verifiable presentation.

session with ACC to upload the river flow information in the form of data verifiable credential. Secure session creation takes 4438 ms on average while uploading data verifiable credential performance is consistent as shown in the Fig. 11. It indicates the consistent performance behaviour of the application despite increasing the load. The time calculation is based on the time consumed for encrypted data verifiable credential creation, its uploading on the ACC using HTTP communication protocol, decryption, and verification of credential before storing in the persistent storage.

- **ACC**: This application is also implemented in Java, web3j and Spring Boot. It is deployed as a microservice on port 9091 which establishes a secure session with clients (i.e., functional units), manages the river flow data and then creates data verifiable presentation on the request of water purification company and a citizen.
- **Water purification company**: Water purification company is also implemented as a client application which requests ACC service to download data of flowID, inflow, and outflow for predicting its future flow of water. It extracts the date of record and presentation from the data verifiable presentation's proof object. The results showed a consistent trend in the ACC service behavior and did not degrade its performance when load was increased.
- **Citizen**: Citizen application is also implemented as a client application which requests ACC service to observe the quality of data and requests quality for checking the quality of water they are consuming. The results in Fig. 11 indicate that the ACC service behaviour is consistent and did not degrade its performance when load was increased.

Further average performance analysis was performed for uploading and downloading data verifiable credentials. Fig. 12 shows promising results and consistent uploading and processing of data verifiable credentials trend i.e., a straight line. It took 1134 ms on average to upload data on ACC. If we compare water purification company and citizen application results for downloading variable credentials, then they are also consistent. On average, water purification company took 1723 ms to download a data verifiable presentation. In contrast, a citizen took 1701 ms. This application took a bit more time because the ACC first creates a data verifiable presentation and then the client applications verify at the attributes level.

The performance results indicate that such a solution is acceptable for smart city applications where real-time or latency is not a critical requirement.

C. SIGNED OFF-THE-SHELF SECURITY FRAMEWORK

SIGNED security stack is built on the modular approach and adopted microservice architecture, which makes it a flexible SIGNED-of-the-shelf security provision framework for a digital twin ecosystem. Furthermore, the verifiable credentials and presentation creation modules are also generic and can be integrated with various applications which have the same functional and security requirements.

VII. CONCLUSION & FUTURE WORK

A novel SIGNED framework is introduced, designed and implemented to answer the research question: 'Can a smart city digital twin be designed and built on verifiably protected and authentic data ownership as well as the ability to perform selective data disclosure to manage the data pipeline from diverse data sources?'. The proof of concept implementation and its application in smart water management use case was verified by using Scyther verification tool. Furthermore, the performance evaluation of the solution is performed to assess the overhead of introducing security measures. The key strengths discussed in the previous section clearly answer the research question that the SIGNED framework can provide a decentralised secure and verifiable data management and selective disclosure approach for a smart city digital twin at a negligible performance cost.

The intrinsic characteristics (e.g., immutability) of blockchain for VDR further strengthens the traceability of credentials. As for SIGNED implementation, no mining is required, which makes the proposed solution environment friendly. However, further research will be needed to apply this solution in a federated digital twins environment, where scalability will need to be further investigated.

The mix-method approach worked well for this work. The proof of concept implementation helped in developing the concrete artefacts to perform the performance evaluation. Furthermore, security verification protocol by using scyther

resulted in testing various possible security attacks. The evaluation results are promising and provide a novel contribution to introduce SIGNED framework in digital twins.

The microservice based design of the SIGNED artefacts provides flexible integration potential with other digital twins. However, for a wider adoption of the SIGNED framework requires further experiments on other use cases from different smart city application domains e.g., mobility, energy, health, planning, etc. This research also paves the way for our future research on secure and verifiable digital assets' marketplace for digital twins.

REFERENCES

[1] D. M. Botín-Sanabria, A.-S. Mihaita, R. E. Peimbert-García, M. A. Ramírez-Moreno, R. A. Ramírez-Mendoza, and J. D. J. Lozoya-Santos, "Digital twin technology challenges and applications: A comprehensive review," *Remote Sens.*, vol. 14, no. 6, p. 1335, Mar. 2022.

[2] M. Singh, E. Fuenmayor, E. Hinchy, Y. Qiao, N. Murray, and D. Devine, "Digital twin: Origin to future," *Appl. Syst. Innov.*, vol. 4, no. 2, p. 36, May 2021.

[3] S. Aheleroff, X. Xu, R. Y. Zhong, and Y. Lu, "Digital twin as a service (DTaaS) in industry 4.0: An architecture reference model," *Adv. Eng. Informat.*, vol. 47, Jan. 2021, Art. no. 101225.

[4] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020.

[5] C. Semeraro, M. Lezoche, H. Panetto, and M. Dassisti, "Digital twin paradigm: A systematic literature review," *Comput. Ind.*, vol. 130, Sep. 2021, Art. no. 103469.

[6] G. Coorey, G. A. Figtree, D. F. Fletcher, and J. Redfern, "The health digital twin: Advancing precision cardiovascular medicine," *Nature Rev. Cardiol.*, vol. 18, no. 12, pp. 803–804, Dec. 2021.

[7] M. N. Kamel Boulos and P. Zhang, "Digital twins: From personalised medicine to precision public health," *J. Personalized Med.*, vol. 11, no. 8, p. 745, Jul. 2021.

[8] R. Sacks, I. Brilakis, E. Pikas, H. S. Xie, and M. Girolami, "Construction with digital twin information systems," *Data-Centric Eng.*, vol. 1, p. e14, Jan. 2020.

[9] A. Nasirahmadi and O. Hensel, "Toward the next generation of digitalization in agriculture based on digital twin paradigm," *Sensors*, vol. 22, no. 2, p. 498, Jan. 2022.

[10] C. Verdouw, B. Tekinerdogan, A. Beulens, and S. Wolfert, "Digital twins in smart farming," *Agricult. Syst.*, vol. 189, Apr. 2021, Art. no. 103046.

[11] G. Mylonas, A. Kalogeras, G. Kalogeras, C. Anagnostopoulos, C. Alexakos, and L. Munoz, "Digital twins from smart manufacturing to smart cities: A survey," *IEEE Access*, vol. 9, pp. 143222–143249, 2021.

[12] F. Laamarti, H. F. Badawi, Y. Ding, F. Arafsha, B. Hafidh, and A. El Saddik, "An ISO/IEEE 11073 standardized digital twin framework for health and well-being in smart cities," *IEEE Access*, vol. 8, pp. 105950–105961, 2020.

[13] L. Deren, Y. Wenbo, and S. Zhenfeng, "Smart city based on digital twins," *Comput. Urban Sci.*, vol. 1, no. 1, p. 4, Dec. 2021.

[14] M. Charitonidou, "Urban scale digital twins in data-driven society: Challenging digital universalism in urban planning decision-making," *Int. J. Architectural Comput.*, vol. 20, no. 2, pp. 238–253, Jun. 2022.

[15] L. Raes, P. Michiels, T. Adolphi, C. Tampere, A. Dalianis, S. McAleer, and P. Kogut, "DUET: A framework for building interoperable and trusted digital twins of smart cities," *IEEE Internet Comput.*, vol. 26, no. 3, pp. 43–50, May 2022.

[16] E. Karaarslan and M. Babiker, "Digital twin security threats and countermeasures: An introduction," in *Proc. Int. Conf. Inf. Secur. Cryptol. (ISCTURKEY)*, Ankara, Turkey, Dec. 2021, pp. 7–11.

[17] D. Popescul and L. D. Radu, "Data security in smart cities: Challenges and solutions," *Inf. Economica*, vol. 20, no. 1, pp. 29–38, Mar. 2016.

[18] Z. Khan, A. G. Abbasi, and Z. Pervez, "Blockchain and edge computing–based architecture for participatory smart city applications," *Concurrency Comput., Pract. Exper.*, vol. 32, no. 12, Jun. 2020, Art. no. e5566.

[19] K. Alshammari, T. Beach, and Y. Rezgui, "Cybersecurity for digital twins in the built environment: Current research and future directions," *J. Inf. Technol. Construct.*, vol. 26, pp. 159–173, Apr. 2021.

[20] Z. Khan, Z. Pervez, and A. G. Abbasi, "Towards a secure service provisioning framework in a smart city environment," *Future Generat. Comput. Syst.*, vol. 77, pp. 112–135, Dec. 2017.

[21] W. Shen, T. Hu, C. Zhang, and S. Ma, "Secure sharing of big digital twin data for smart manufacturing based on blockchain," *J. Manuf. Syst.*, vol. 61, pp. 338–350, Oct. 2021.

[22] D. Lee, S. H. Lee, N. Masoud, M. S. Krishnan, and V. C. Li, "Integrated digital twin and blockchain framework to support accountable information sharing in construction projects," *Autom. Construct.*, vol. 127, Jul. 2021, Art. no. 103688.

[23] B. Putz, M. Dietz, P. Empl, and G. Pernul, "EtherTwin: Blockchain-based secure digital twin information management," *Inf. Process. Manage.*, vol. 58, no. 1, Jan. 2021, Art. no. 102425.

[24] S. Suhail, R. Hussain, R. Jurdak, and C. S. Hong, "Trustworthy digital twins in the industrial Internet of Things with blockchain," *IEEE Internet Comput.*, vol. 26, no. 3, pp. 58–67, May 2022.

[25] J. Kendall, "National digital twin: Integration architecture pattern and principles," 2021. [Online]. Available: https://www.cdbb.cam.ac.uk/files/architecture_principles_final.pdf and https://www.repository.cam.ac.uk/handle/1810/321334, doi: 10.17863/CAM.68207.

[26] *Self-Sovereign Identity and IoT*, Sovrin Foundation, Provo, UT, USA, Nov. 2021.

[27] A. G. Abbasi and Z. Khan, "VeidBlock: Verifiable identity using blockchain and ledger in a software defined network," in *Proc. Companion 10th Int. Conf. Utility Cloud Comput.*, New York, NY, USA, Dec. 2017, pp. 173–179.

[28] *Personal Identification—ISO-Compliant Driving Licence—Part 5: Mobile Driving Licence (mDL) Application*, Standard ISO/IEC 18013-5:2021, ISO, 2021.

[29] *Java*. Accessed: Nov. 25, 2022. [Online]. Available: https://www.java.com/en/

[30] *Springboot*. Accessed: Oct. 19, 2022. [Online]. Available: https://spring.io/projects/spring-boot

[31] *Web3J*. Accessed: Nov. 22, 2022. [Online]. Available: https://docs.web3j.io/4.8.7/

[32] *Ganache*. Accessed: Nov. 22, 2022. [Online]. Available: https://next-stack.github.io/ganache/

[33] *Ethereum*. Accessed: Nov. 22, 2022. [Online]. Available: https://ethereum.org/en/

[34] *Abi*. [Online]. Available: https://www.blockchain-council.org/solidity/solidity-abi-application-binary-interface/

[35] *Truffle*. Accessed: Oct. 21, 2022. [Online]. Available: https://github.com/trufflesuite/truffle

[36] C. Cremers, "The scyther-proof security protocol verification tool," ETH Zurich, Zurich, Switzerland, Tech. Rep., Sep. 2022. [Online]. Available: https://people.inf.ethz.ch/basin/pubs/meier10.pdf

ZEESHAN PERVEZ (Senior Member, IEEE) is currently a Professor of computer science with the University of the West of Scotland (UWS). He is an ACM Distinguished Speaker, a Senior Fellow of the Higher Education Academy (U.K.), and a Full Member of EPSRC Peer-Review College (U.K.). He has a strong track record of securing research, industry, and capacity-building funding from the Innovate U.K., Scottish Funding Council, Innovation Centers, European Commission, Erasmus+, Microsoft Research, and international research institutes. He has over 18 years of research and industry experience in addressing technological and societal challenges and designing and developing enterprise-ready solutions. He has outstanding performance and success in securing substantial research funding, enterprise engagement, high-quality research outputs, and associated wider research/societal impact. His areas of expertise are data science, applied ML/AI, cybersecurity, Industry 4.0, and edge/fog computing. The application areas of his research include, but are not limited to, Industry 4.0, smart cities, social housing, predictive maintenance, facilities management, healthcare, data science, and ICT4D. He has been regularly invited by national and international research and academic institutes to deliver talks on topics ranging from the Internet of Things to data science, cybersecurity, and cloud computing.

He has served as a TPC member for more than 100 conferences and a regular reviewer for more than 30 IEEE TRANSACTIONS, Elsevier, and Springer journals. He has successfully supervised to completion: 29 postdoctoral fellows, postgraduate research/taught, and undergraduate students (as lead and a co-supervisor). He won the 2019 UWS STARS Award—Staff Appreciation and Recognition Scheme. He has chaired/evaluated over 28 doctoral exams for the U.K. and international higher education institutes.

ZAHEER KHAN received the B.Sc. degree in computer science, the M.S. degree in information technology, and the Ph.D. degree in computer science. He has over 18 years of academic research experience. He is currently a Professor of computer science with the University of the West of England (UWE), Bristol, U.K. His primary research addresses intelligent and secure distributed systems, including developing and applying new software processes and methods for systems development by using modern technologies such as blockchain, the IoT, digital twin, cloud/edge computing, and big-data management and analysis. He has been applying his research on digital urban governance and sustainable smart future cities by engaging with many European cities through different international projects, since 2009. He also looks after the smart cities theme of the Computer Science Research Centre (CSRC), UWE. He has extensive experience working on several externally funded large-scale collaborative research and innovation projects. These projects include FP6 Health-e-Child (integrated health informatics), FP6 HUMBOLDT (geospatial data harmonization), FP7 LifeWatch (biodiversity digital infrastructure—preparatory phase), FP7 Urban API as a Co-PI (digital participatory governance through 2-D/3-D visualization/simulation tools), FP7 DECUMANUS as a Co-PI (remote sensing for environmental and climate change applications), H2020 Smarticipate as the UWE Project Manager (open data, 2-D/3-D visualization, and digital participatory governance platforms), H2020 CURE Co-PI (remote sensing for cross-thematic applications for urban resilience) Project, and Horizon Europe GREENGAGE Project as the UWE Project Manager (citizen observatories, citizen science, earth observation, and decision making for climate change adaptation and mitigation). He is also on the scientific committee of several international conferences and journals.

ABDUL GHAFOOR (Member, IEEE) received the M.S. degree in network technologies from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2004, and the Ph.D. degree in network security from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2011. He has more than 14 years of research, development, and teaching experience in information security and software development. He is a certified Information Design Assurance Red Team (IDART) from the Sandia National Laboratories, Albuquerque, New Mexico, and the Monterey Institute of International Studies, Monterey, CA, USA. He is the main architect of CryptoNET and a senior architect with the SAGE Design Team. He is currently a Research Scientist with RISE Research Institutes of Sweden AB (RISE), Kista Office, in the domain of trusted decentralized systems, digital identities, and zero-knowledge proof. Before joining RISE, he was part of a research team with Communication Systems, KTH, as an Associate Research Engineer and a Visiting Faculty Member. He served as a Faculty Member with Federal Education Directorate (2002–2004), KUST (2004–2006), and NUST (2006–2007 and 2011–March 2015). He also worked as a Security Consultant with SETECS, Inc., for five years. He has published more than 35 research papers and scientific articles in international journals and conferences. His research interests include network security, blockchain technologies, self-sovereign identities, zero-trust architecture, and secure applications for smartphones. He actively participates as a reviewing committee member of several reputed journals and conferences.

KAMRAN SOOMRO (Member, IEEE) received the bachelor's and Ph.D. degrees in computer science and has over ten years of experience in research and academia.

He is currently a Senior Lecturer in computer science with the School of Computing and Creative Technologies, University of the West of England, Bristol, U.K. He has expertise in distributed systems such as grids and clouds and technologies such as Hadoop, spark, and storm clusters among other distributed technologies. He has also been involved in various externally funded research projects such as EU FP7 neuGRID, urbanAPI, DECUMANUS, Horizon 2020 Smarticipate, CURE, and Horizon Europe GREENGAGE Project. He has also worked on a collaborative project funded under the British Council Newton Institution Links Programme with Taylors' University, Malaysia. He has also led a KTP involving machine learning as well as other research projects in the same area. His research interests include the use of ICT technologies for smart cities, urban management and healthcare, knowledge management, artificial intelligence, big data, and natural language processing.