IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 27 April 2023, accepted 15 May 2023, date of publication 29 May 2023, date of current version 8 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280928

RESEARCH ARTICLE

Security-Aware Provenance for Transparency in IoT Data Propagation

FARIHA TASMIN JAIGIRDAR[1], **(Member, IEEE), BOYU TAN**[2], **CARSTEN RUDOLPH**[1], **AND CHRIS BAIN**[1]

[1]Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia
[2]China Mobile Group Design Institute Company Ltd., Beijing 100080, China

Corresponding author: Fariha Tasmin Jaigirdar (fariha.jaigirdar@monash.edu)

ABSTRACT A successful application of an Internet of Things (IoT) based network depends on the accurate and successful delivery of data collected from numerous sources. A significant concern in IoT systems arises when end-users do not have sufficient transparency and are unaware of any potential data manipulation and risk in each step involved in data propagation. One potential solution is to integrate security metadata in IoT-based security-aware provenance graphs that provides better transparency with security awareness at each step of data propagation. In this paper, we integrate security metadata into the provenance graph with predefined security policies. We design a hypothetical IoT-Health scenario with possible threats: node cloning, fault packet injection, denial of service, unauthorized access, and malicious code injection. We simulate these threats in six cases to identify relevant risks. Our findings show how a security-aware provenance graph can offer end users greater transparency and security awareness by identifying failed signature verification (case 1), denial of service (case 2), unauthorized access (case 3), intrusion detection (case 4), missing WAF (case 5), and permission violation (case 6). We evaluate the transparency through obtaining authentication, integrity, availability and detecting underlying threats. Accordingly, this study promotes better risk assessment and decision-making for users with negligible performance overhead.

INDEX TERMS Internet of Things (IoT), data provenance, IoT-Health, transparency, security-awareness.

I. INTRODUCTION

Growing innovations in the Internet of Things (IoT) technology create opportunities for future visions of smart cities, smart healthcare, smart transportation, and a smart world [1], [2], [3]. These are the prospects for future business investments into new applications all over the world. In IoT-based environments, different heterogeneous components are expected to be independent participants and communicate with each other without human intervention [4]. This intelligent feature of multi-layer IoT architecture promotes productivity, usability, and simplifies the use of IoT systems. Also, the increasing number of interoperable protocols and smart entities breaks the traditional technical barriers and enables novel IoT-based solutions available to end-users without the need to have strong technical knowledge of the systems involved [4]

IoT-based environments with heterogeneous components bring more security concerns compared with the traditional internet environment [5]. Different security threats and privacy concerns have been summarized by researchers [2], [4], [6], [7]. To combat these, studies have proposed novel mechanisms, including improved advanced encryption standards (AES) and quantum walks (QW) to secure data transmission in IoT networks [8], [9]. Nevertheless, it can be difficult to verify that every component of multi-layer IoT architectures maintains an appropriate level of security deployment as there is no end-to-end security that enables a complete validation of the security mechanisms used in the overall structure [10].

Moreover, as soon as attacks can have serious consequences to human life or create significant financial damage, it becomes a major concern that end-users are not able to perceive any potential risks or attacks [11], [12] and

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

end-users are not able to estimate or interpret if the data they see is trustworthy [10]. Thus, users do not have suitable cyber-situational awareness [13], [14] to know whether cyber-attacks are possible or have even occurred while the data was propagating through IoT systems. As a result, confirming trustworthiness by improving transparency and security awareness of IoT systems, particularly regarding security risks, has become a new demand in this domain [11], [15].

Encouraged by the above discussion, in this paper, we aim to provide transparency in IoT data propagation by identifying relevant risks using a security-aware data provenance-based approach [16]. Different studies have been carried out to explore the relationship between data provenance and IoT networks [17], [18], [19]. Among these, research on 'secure provenance data' highlights the security of provenance records and/or provenance graphs themselves [20], [21], [22], and lacks exploring transparency from security awareness perspectives. We argue that during data propagation, security relevant information is crucial to provide security status. Documenting this information along with data lineage history (data provenance) would provide transparency from security viewpoints. To extend data provenance graphs with security information, the authors proposed a novel security-aware IoT provenance model named Prov-IoT in [16]. It includes evidence on security-aware features (denoted as security metadata) in data provenance graphs considering all processing steps of data propagation.

The Prov-IoT model and an associated class diagram provide a framework for building a suitable provenance graph with security metadata, relevant granularity policies, data-subject attribution policies, and provenance validation. This model is the first theoretical foundation to connect a data provenance graph with security awareness. However, its operation in real-life IoT scenarios and practical effects on system transparency during data propagation are uncertain. Moreover, how different security policies are helpful as fault/attack detection principles and how security-aware provenance graphs are validated in a practical setup remains unknown. Therefore, in this paper, we consider the Prov-IoT model for further analysis in identifying security threats in a designed IoT-Health architecture through a proof-of-concept based implementation using six threat-based cases. Specific contributions of this research are as follows.

1) We build a hypothetical IoT-health scenario as the basis of the case study. We generate a security-aware provenance graph for the designed scenario and present novel techniques, including pre-defined security policies, graph identifiers, and operation schemes for the graph.
2) We simulate the process and conduct implementation using Alibaba ECS cloud virtual machines and the Neo4j graph database. We design and simulate attacks (node cloning attack, fault packets injection attack, denial of service attack, an unauthorized access attack, and malicious code injection attack) in six case studies to misuse some vulnerabilities. Further, we provide the threat detection principles for these six case studies.
3) Finally, we evaluate the implementation in terms of transparency and conduct performance analysis.

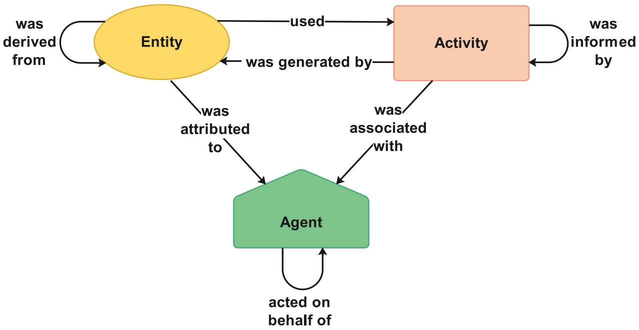


FIGURE 1. Basic PROV-DM model.

The rest of the paper is organized as follows: Section II discusses the background and motivation of the work. We demonstrate an IoT-Health case study and associated threats in section III. Section IV illustrates the construction and operation of the security-aware provenance graph for the designed case-study. Experiments on six cases and results are described in section V. We evaluate the transparency of the security-aware provenance graph along with an analysis on system's performance in Section VI. Section VII projects the potential implications and limitations of the study. Finally, we conclude the paper with future works in Section VIII.

II. BACKGROUND, RELATED LITERATURE AND MOTIVATION

This section introduces relevant background knowledge and clarifies the motivation to conduct this research.

A. DATA PROVENANCE

The concept of provenance came from the area of art long time ago. It refers to a document that records the source along with every owner of an artwork [23]. The purpose of collecting provenance information is to guarantee the authenticity of every masterpiece. The Open Provenance Model (OPM) defines the preliminary provenance description and demonstrates the provenance graph [23].

A standard provenance data model (named PROV-DM) adopted by W3C is illustrated in Fig. 1. The PROV-DM is a refinement of the OPM, and covers a broader range of application domain. It structures the whole system into three elements along with relationships among them. To illustrate the model in an IoT scenario, the entity represents the data transformed within the system. The activity is the action that causes the change (processing, modifying etc.) of that data. The agent triggers the action and takes responsibilities for it. Establishing a standard provenance graph for the hypothetical IoT scenario is the initial step of achieving the security-aware provenance graph in our implementation. In the following subsections, we provide a systematic discussion of existing

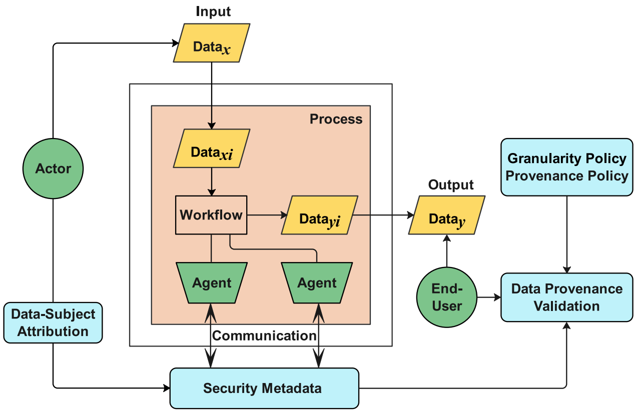


FIGURE 2. Prov-IoT model [16].

literature on secure provenance and IoT transparency to project the current research gap and demonstrate this paper's motivation.

1) SECURE PROVENANCE

Research in secure provenance can be classified into three areas: 1) Security of provenance record, 2) Authenticity of source of the data in provenance records, and 3) Security-aware provenance (including security metadata at all steps involved in data propagation).

The first area includes different security mechanisms to *secure provenance graphs/provenance records* themselves, ranging from digital signatures on hash trees to blockchain-based approaches [21] and [24]. To illustrate some, Baracaldo et al. propose a framework on access control based blockchain technology to protect data provenance records [25]. A secure data forensic provenance scheme in cloud computing is proposed by Li et al. [26]. Shams et al. in [27] present a trustworthy and efficient provenance management mechanism in the cloud, named SECProv. In a recent study, Shreya et al. introduce a secure decentralized application framework for sharing files and data provenance [28]. In [29], a theory about abstracting provenance graphs is introduced to protect sensitive information from data providers, when provenance data sharing happens in ad hoc collaborative partnerships. Moreover, Sigwart et al. discuss how introducing blockchain technology with data provenance can make data more secured [30].

The second area of research deals with identifying the *authenticity of the source of the data in provenance records*. For example, Aman et al. propose secure data provenance in IoTs in the form of a lightweight IoT protocol [17]. It uses Physical Unclonable Functions (PUFs) to provide physical security and uniquely identify an IoT device. In addition to practicalities with validating PUFs, integration into the different phases of data propagation and various layers of the IoT environment are not addressed. Use of blockchain technology to trace provenance in establishing social trust is discussed in [31]. This work defines how a technical system for tracing origins, ownership and authenticity can transform social trust using two case studies. Since IoT devices are physically exposed and heterogeneous in nature, research identifies device registration and data generation to be the most vulnerable phase [32]. Accordingly, many researchers discuss authentication approaches (for example, hashing with element extraction, secure key establishment using elliptic curve cryptography, lightweight authentication protocol for securing RFID tags) in the initial phases of data propagation [33]. These methods can provide an initial root of trust for origin of authentication, but relating this trust to the end-user remains open.

However, these two areas of research do not consider the data processing and fusion among different layers, cross-layer dependency, and step-wise data propagation in IoT environments. Including security-aware properties at each step of data propagation offers end-users more comprehensive IoT transparency as it projects the security status/behavior of device/sensor(s) in an IoT architecture, software running on the devices, data processing mechanisms, as well as communication channels' properties. Therefore, we work on the third area of secure provenance, which presents *security-aware provenance*, including security metadata. The only existing theory regarding the security-aware provenance model is the data provenance graph with security metadata named Prov-IoT model [16]. Fig. 2 presents the Prov-IoT model, the earlier works of the authors. It includes necessary attributes for step-wise data propagation and adds security-aware functions (security metadata, data-subject-attribution, policies, data provenance validation) to store each information from data generation to visualization through processing. Although this framework demonstrates a security-aware IoT provenance architecture, it relies more on the theory construction and leaves an opportunity for real-life implementation along with acceptability analysis. Therefore, this paper emphasizes on its usability and feasibility study with a user-friendly function to bring tangible comprehensive transparency for IoT end-users via security-aware provenance graphs.

2) RELATED TECHNIQUES

There are several comparable approaches to enhancing system transparency via security assessment, which encompass attack tree, state transition diagram, and attack graph [34], [35], [36]. For example, authors proposed in [34] an attack tree model to provide a systematic representation of attack scenarios and address security issues through quantitative evaluation. However, attack trees do not possess the ability to recognize how attacks and defenses interact within a system. Another representative method, the state transition diagram, is adopted in [35]. This project aims at identifying the vulnerabilities in the firmware trust verification procedure in combination with the state transition diagram. But this technique is suited for explaining the behavior of a single object and inherently has difficulty describing activity in an IoT system that involves several objects. Sahay et al. in [36] construct the attack graph to prevent the exploitation of the vulnerabilities in an IoT network. Attack graph is capable of modeling the potential pathways that attackers use to penetrate a network but is unable to depict data transition and propagation processes within an architecture, which is crucial in the context of particular IoT environments such as medical care and autonomous driving.

Consequently, the security-aware provenance graph presented in this work incorporates security metadata that bridges the connections between attacks and defenses in the IoT system. Additionally, the provenance model can be built on multiple entities, which clearly demonstrates their relations associated with one behavior. Finally, the provenance graph supports the visualization of transitions in the data lifecycle, which is of great importance for IoT security since it can locate (possibly) compromised data. The above discussion explains why the security-aware provenance graph outperforms alternative techniques in delivering transparency for IoT scenarios.

3) PROVENANCE AND IoT TRANSPARENCY

Several theories and/or methodologies based on provenance techniques have been proposed to increase IoT transparency, some focusing on traceability, others on accountability [37], [38], [39], [40], [41], [42]. In [37], a software infrastructure based on W3C provenance recommendations is established to store a sensor environment's security-related data. End-users can specify privacy preferences to get notifications of possible risks and violations. This project improves users' control over IoT systems' transparency and guarantees data trustworthiness before using the data. However, it does not consider step-wise data propagation information and only simulates a simple scenario with a smartphone application; its scalability and security-aware properties are uncertain.

Based on the PROV-DM model, a provenance collection framework is created for IoT devices in [39]. This work integrates provenance and IoT scenarios to provide provenance collection and provenance check functions. Besides, a database is deployed in the cloud to store all the provenance data, which improves scalability and compatibility. Nevertheless, no implementation is conducted to test the performance of this model nor step-wise security information is documented by the proposed method. Moreover, provenance checking merely shows if an IoT device complies with rules, but it cannot perceive potential cyber-attacks or system vulnerabilities. Therefore, it is hard for this level of IoT transparency to provide comprehensive and valid information to end-users. A provenance technique that enhances IoT traceability with visualization is proposed in [40]. It verifies the correctness of data propagation by determining linkability and unlinkability between IoT nodes. However, this plan also faces the challenge of how to supply evidence to make users conduct forensic tasks related to attacks. Apart from models and schemes, algorithms for IoT provenance verification is introduced in [41]. Nevertheless, it is not achievable to run complex algorithms in an IoT edge node due to its resource-constraint characteristics.

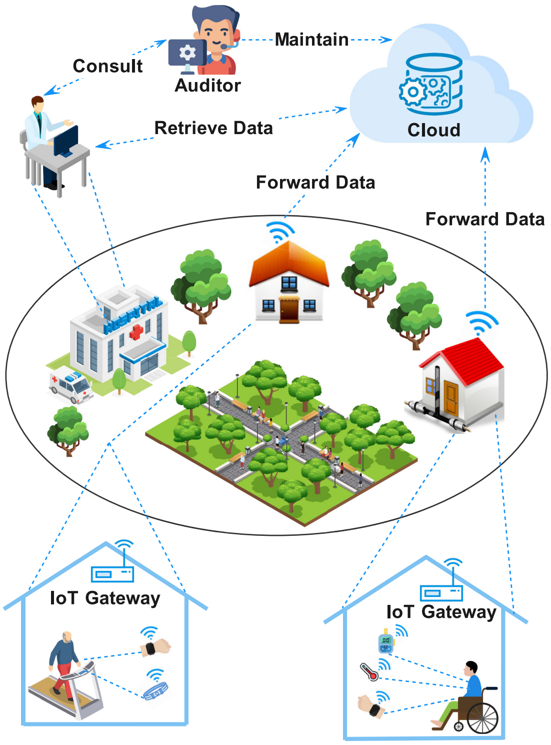


FIGURE 3. A hypothetical IoT-health scenario.

To sum up, models or schemes for improving IoT transparency proposed by the above studies either provide low-level transparency or lack experiments to validate their real IoT scenarios' feasibility. Additionally, transparency through security awareness is a question that has previously never been addressed because existing provenance models are unable to preserve security-related evidence of IoT systems. In our scheme, instead of only being conscious of violations, we work on security-aware provenance graphs for IoT consumers to obtain security evidence and understandable validation results. From external attacks to internal risks, it senses them, locates them, and eventually gives standardized security evidence to make appropriate decisions. Most significantly, we consider usability and achieve automated validation of graphs, which is beneficial for end-users' decision making. Therefore, our implementation demonstrates better applicability for real-life IoT scenarios except providing more comprehensive transparency.

III. CASE STUDY AND THREAT MODEL

This section presents a hypothetical IoT-based healthcare scenario as the background for the case study, then elaborates possible threats related to this IoT system.

A. A HYPOTHETICAL IoT-HEALTH SCENARIO

This hypothetical case study is based on two elderly people, Tom and Jim. They suffer from Type 2 diabetes and hypertensive disease, respectively. Chronic diseases are controllable as long as their conditions are monitored and they receive doctor's instructions and comply with those in time [43]. A designed IoT-Health architecture for such a scenario is projected in Figure 3.

In the sensing layer, each patient wears one or more IoT devices designed for various health-monitoring tasks. Monitoring applications run in the IoT gadgets to fetch raw medical data from sensors placed on a patient's body. In the network and data processing layer, smart IoT gateways (deployed in the patients' homes) and the cloud server (provides computing services) are essential components. The IoT gateway pre-processes and aggregates captured data and transmits this data to the cloud. Meanwhile, it undertakes device management tasks such as device registration and device control. After the cloud service identifies and processes the data received from the gateway, processed data and potential results from automated data analysis are stored in the cloud. In the application layer, the doctor retrieves patients' data, analyses results from the cloud via a web application using his computer in the hospital, and decides on treatments. We assume that the doctor usually has no access to the original sensor data. However, if there is an indication of an increased risk of data being manipulated or insecure systems being involved, the doctor might need to react by adding additional checks potentially based on the original data. Accordingly, a technician, indicated in the system architecture as an auditor, monitors the system, provides technical assistance to the doctor, and regularly updates or maintains the system.

B. THREAT MODEL

Attackers in a heterogenous IoT environment can manipulate information, insert fake information and violate data integrity. They can also violate the authenticity by exploiting a weak security deployment to intrude into the end-user's device and modify the availability of IoT data at different steps of data propagation. Hence, it may cause further damage to the IoT system or the trustworthiness of IoT data. We summarize possible threats for the designed IoT scenario in the following paragraphs.

1) POSSIBLE THREATS

- **Node cloning and Injecting fraudulent packets**. In a node cloning attack, an attacker can copy an existing node's identifier and violates the node's authenticity within the system.
  Moreover, an adversary can insert fraudulent packets into communication traffic between IoT gateway and sensors/smart watches [44]. This attack is possible in two different ways: insertion and manipulation. In insertion attacks, the malicious user may generate packets that seem legitimate and insert them into the traffic. Manipulation attacks include capturing packets, altering information within them, and violating data integrity. In order to identify these types of attacks, security evidence should demonstrate the implementation of secure communication protocols.
- **Denial of service (DoS)**. With resource constraint IoT nodes, for example, smartwatches in this scenario, it is possible to launch DoS attacks [45]. Hence, the attackers may take advantage of a smartwatch's low battery status, aim to violate the availability of IoT services or leverage this situation to create more attacks. Although DoS attacks can not alter data, the provenance graph must monitor data timing if emergency reactions depend on the timely reporting of the values.
- **Unauthorised access**. Attackers can gain unauthorized access by accessing the cloud API in the given scenario. API-level attacks may allow attackers to compromise the API and get cloud services even if their identities do not meet the access control criteria [46]. Aside from the possibility of data leaking, the system's integrity is also jeopardized. Thus, security controls need to include application-level mechanisms to prevent manipulations on the API level. In addition to the cloud service, unauthorized access to the smart IoT gateway has an impact on privacy because it governs and maintains device management information.
- **Malicious code injection**. Since an end-user may access different web applications, a malicious code injection attack is very realistic in the designed scenario. An end-user accesses processed data from the cloud using different web applications, and the application itself can become a target of those malicious codes. For example, if the application version is outdated or the end-user's device has a weak security control deployment, it is easy for malicious functions to gain higher permissions and then capture private information from end-users' devices [47].

Given the possibility of security threats at different layers of data propagation, ensuring the accuracy, transparency, and timeliness of the data transferred throughout the system is crucial. Therefore, along with data lineage, including security-relevant information for devices and communication steps allows continuous data auditing and security awareness throughout data propagation.

IV. SYSTEM MODEL: A SECURITY-AWARE PROVENANCE GRAPH

This section illustrates the construction and operation of the security-aware provenance graph.

A. GRAPH CONSTRUCTION

The construction of a security-aware provenance graph consists of initializing the graph structure with necessary attributes and adding new attributes and relationships to the graph.

1) INITIALIZATION

According to the standardized PROV-DM model published by W3C, a general provenance graph for the designed IoT-Health scenario can be constructed as illustrated in Fig. 4.

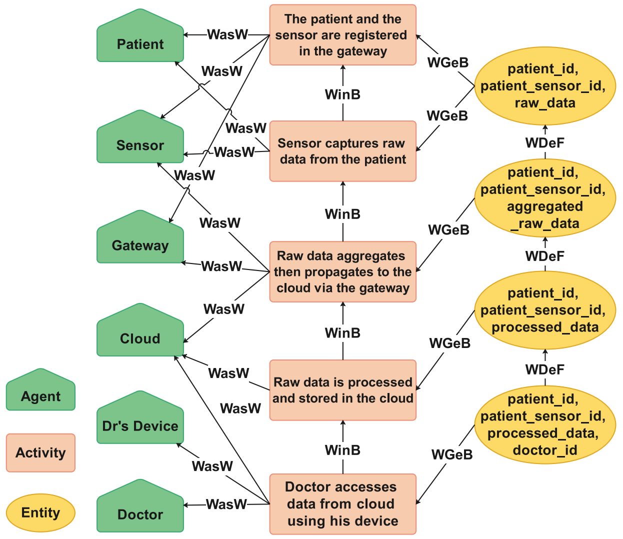

Patient
Sensor
Gateway
Cloud
Dr's Device
Doctor
Agent
Activity
Entity
The patient and the sensor are registered in the gateway
Sensor captures raw data from the patient
Raw data aggregates then propagates to the cloud via the gateway
Raw data is processed and stored in the cloud
Doctor accesses data from cloud using his device
patient_id, patient_sensor_id, raw_data
patient_id, patient_sensor_id, aggregated _raw_data
patient_id, patient_sensor_id, processed_data
patient_id, patient_sensor_id, processed_data, doctor_id
WasW
WinB
WGeB
WDeF

FIGURE 4. **A general provenance graph for the IoT-health scenario.**

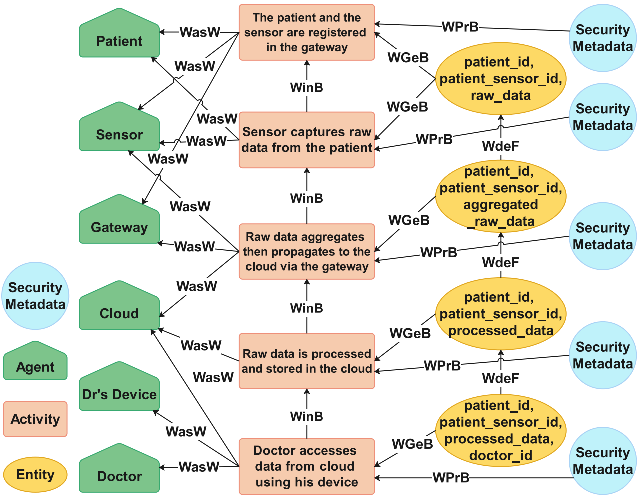

Patient
Sensor
Gateway
Cloud
Dr's Device
Doctor
Agent
Activity
Entity
Security Metadata
The patient and the sensor are registered in the gateway
Sensor captures raw data from the patient
Raw data aggregates then propagates to the cloud via the gateway
Raw data is processed and stored in the cloud
Doctor accesses data from cloud using his device
patient_id, patient_sensor_id, raw_data
patient_id, patient_sensor_id, aggregated _raw_data
patient_id, patient_sensor_id, processed_data
patient_id, patient_sensor_id, processed_data, doctor_id
WasW
WinB
WGeB
WDeF
WPrB

FIGURE 5. **A security-aware provenance graph for the IoT-health scenario.**

The whole scenario is depicted by three main node types: agent, activity, and entity [23]. Relationships among these three are portrayed as WGeB (Was Generated By), WDeF (Was Derived From), WInB (Was Informed By), and WAsW (Was Associated With) [23].

The first activity is registration, which is accomplished by three agents (patient, sensor, gateway). Data capturing is the second activity in the sensing layer, where two agents (patient and sensor) trigger this activity. These two activities produce the first entity (raw data, sensor ID, and patient ID). The third activity describes data propagation in the network layer, which is associated with the sensor, gateway, and cloud. This activity creates the second entity (aggregated raw data along with sensor ID and patient ID). The cloud agent conducts the fourth activity (raw data is processed and stored in the cloud). At the same time, a new entity (processed data along with sensor ID and patient ID) is created. The final activity is data retrieval in the application layer. It is conducted by three agents (doctor, doctor's device, cloud), and the final entity (processed data along with sensor ID, patient ID, and doctor ID) is generated by this process.

2) NEW ATTRIBUTE AND RELATIONSHIP

The general provenance graph (presented above) represents the standard flow of data, which describes all the processes and how they interconnect with each other in the system. However, without security-relevant information in the provenance graph, activities during data propagation are not transparent to the end users, and they cannot detect/discover inconsistent parts, if any. Therefore, we extend the fundamental structure of the PROV-DM model.

Based on the theory presented in [16], as shown in Fig. 5, we use a new attribute named security metadata to store security evidence involved with each activity. The security metadata records security attributes such as active security controls, security protocols, operating system versions, and/or authentication reports. Also, we put forward a new relationship between security metadata and activities called WPrB (Was Proved By). It indicates the presence of security controls for each activity, which can be proved by its security metadata. The validation result of inspecting security metadata can provide positive or negative indicators. Hence, when an activity is confirmed to be conducted successfully and securely, the entities and agents connected with that activity are proven to satisfy particular requirements. Meanwhile, suspicious activities weakening security controls or the existence of vulnerabilities due to the lack of security controls can be identified directly under this structure. In contrast to the original PROV-DM, this structure expands users' cognitive understanding of the security context of IoT systems and further promotes system transparency by linking security evidence with IoT architecture.

B. GRAPH OPERATION

This section explores how the security-aware provenance graph operates in the designed IoT-Health scenario with predefined security policies and graph validation.

1) PRE-DEFINED SECURITY POLICIES

In order to inspect security metadata, appropriate security policies are critical. However, various challenges lie in developing proper security policies for IoT environments. First, there is usually no end-to-end security relation in IoT scenarios. Thus, straightforward solutions, such as end-to-end encryption, cannot be applied. Second, since different IoT scenarios vary greatly, there is no unified security standard that suits all cases equally [48]. Therefore, we have to create security policies by a comprehensive consideration of deployed protocols, security settings, hardware, and software information within an IoT environment. System transparency can be better projected by inspecting such evidence stored in each security metadata node and locating specific insecure parts.

For the designed IoT scenario, we specify requirements for each activity with information related to hardware, software, service providers, protocols, time slots, and security mechanisms (see Table 1). The corresponding security metadata is examined to demonstrate an activity's security state or security awareness. If all required security controls are operational and no malicious activity has interfered with them, the same information should be displayed in the associated security metadata node. Differences, on the other hand, indicate the presence of rule violations or risks.

2) GRAPH VALIDATION

The content displayed in a security-aware provenance graph can be validated after explicit policies for security controls are defined. However, numerous data sources exist in an IoT scenario and transmit data periodically. During the interim between each validation, a number of graphs with potentially a large amount of data in each graph can be formed. Manual validation is error-prone and time-consuming under these circumstances. Therefore, we design a method for end-users to readily acquire validation findings.

a: GRAPH IDENTIFIER

To provide rapid graph validation while taking into account the intricate interactions between IoT sensors, provenance graphs, and time, we first create a unique identity for each graph. For the designed scenario, we set a graph identifier combined by PatientId, DeviceId, and the time when processed data is generated at cloud. Therefore, provenance graphs can be classified and identified by information inside identifiers such as PatientId or DeviceId. It is worth mentioning that the construction of graph identifiers can vary for different IoT scenarios. Real-life scenarios may require more information about devices and time to differentiate graphs.

b: VALIDATION SCHEME

As illustrated in Fig. 6, a graph database is deployed to store security-aware provenance graphs, where each graph maintains a unique identifier. The validation method contains a graph identifier search function, validation function, and security policies (Full details are outlined in Algorithm 1). It also returns high-level validation results that are easy to understand. The auditor is authorized to update the content of security policies or adjust the number of policies if necessary. Beyond that, he can access the graph database to drill down on the details and conduct audit tasks directly.

V. IMPLEMENTATION

This section illustrates the system setup and six experiments we conducted to demonstrate how the security-aware provenance graph improves system transparency.

A. SYSTEM SETUP

The implementation method is illustrated in Figure 7. Initially, each IoT device D_j needs to establish public key infrastructure (PKI) and register in its IoT gateway G_i. After that, D_j packets raw data and security metadata, and sends raw data packets along with digital signature to G_i. G_i verifies data packets and aggregates raw data from data sources. We set the gateway to transmit aggregated data packets to the cloud C as long as it receives ten raw data packets from a device. Apart from health data, aggregated data packets also comprise security metadata of D_j and G_i. Once the cloud C receives packets and successfully decrypts them via pre-shared keys negotiated with G_i, it processes data by visualization techniques. Next, the cloud names data charts using the time

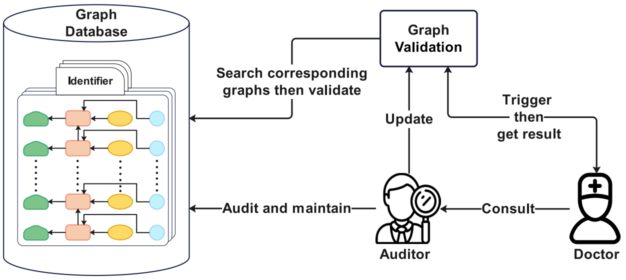


FIGURE 6. A scheme to achieve validation of security-aware provenance graphs.

Algorithm 1 Validation of Security-Aware Provenance Graphs

```
function SrchGids(deviceId) Step 1 search graph identifies that contain
specific deviceId
Step 2 return a list contains identifiers found in
step1
end function

function SectyMetadataVal(identifier)
    Step 1 find the graph via its identifier then validate each
security metadata node according to pre-defined security policies
Step 2 return specific processed data and activities with inconsistent records
end function

Input: DeviceId $Dx$
Output: a list $Li$ of unreliable data
  $(Li, Lp) \leftarrow (empty\ list, empty\ list)$
  $Lp \leftarrow$ SrchGids($Dx$)
  $I \leftarrow$ LENGTH($Lp$)
  for $k \leftarrow 1$ to $I$ do
      $Li.insert$(SectyMetadataVal($Lp[k]$))
  end for
  if LENGTH($Li$) == 0 then
      return No unreliable record
  else
      return Li
  end if
```

TABLE 1. Pre-defined policies as a standardization.

Activities	Pre-defined policies for the patient Tom	Pre-defined policies for the patient Jim
Activity1	patient: Tom, sensor: eHealth watch, IoT gateway: smart gateway(001), register time:	patient: Jim, sensor: Health Monitor, IoT gateway: smart gateway(002), register time:
Activity2	wireless module: 802.11b/g/n 2.4 GHz, module: eHealth watch, OS: wear os, auth: 2FA, app name: sugar monitor, average power consumption: <0.1w/h	wireless module: 802.11b/g/n 2.4 GHz, module: Health Monitor, OS: Tizen, auth: fingerprint identification, app name: smart blood pressure, average power consumption: <0.1w/h
Activity3	IoT device signature verification: EC-prime256v1-SHA256 verified, encryption method: AES-256-CBC security protocol: TLS-RSA-AES-256SHA state: No intrusion is detected	IoT device signature verification: RSA2048-SHA512 verified, encryption method: AES-256-CBC security protocol: TLS-RSA-AES-256SHA state: No intrusion is detected
Activity4	service provider: Alibaba ,data status in the cloud: unencrypted	service provider: Alibaba ,data status in the cloud: unencrypted
Activity5	OS: windows10, browser: Google Chrome(version83.0.4103.97), WAF: Mod security, app: eHealth monitor(web application with least privilege), Dr identity verification: RSA1024-SHA256 verified, security protocol: TLS-RSA-AES-256SHA, retrieve time:	OS: windows10, browser: Google Chrome(version83.0.4103.97), WAF: Mod security, app: eHealth monitor(web application with least privilege), Dr identity verification: RSA1024-SHA256 verified, security protocol: TLS-RSA-AES-256SHA, retrieve time:

they are generated and stores charts according to DeviceID. Meanwhile, C produces security-aware provenance graphs. Finally, the doctor Dr sends retrieve requests to the cloud, and the cloud can justify the data retriever's identity. This action is also recorded in the corresponding provenance graph.

We assume the CSP (cloud service provider) is trusted and does not deliberately violate this IoT system's data trustworthiness. While this is a rather strong assumption, it is obviously needed in this scenario. Weakening this assumption would require some end-to-end security relationship between the sensors or gateway and the doctor. However, identifying this requirement emphasizes the fact that a malicious cloud provider may simply modify the data and computation at will in many applications. It should be noted that the trustworthiness of the cloud provider does not imply that all required security controls are always in place. Thus, it is still necessary to report active security controls as metadata in the provenance graph.

Based on the implementation method for the IoT-Health scenario showed in Fig. 7, we use the Alibaba ECS virtual machine, a ubuntu-bionic-18.04-amd64-server with 4GB memory and two virtual cores CPU, as the cloud server. We use Neo4j, which is a highly-scalable and user-friendly graph database to store provenance graphs. The Neo4j retrieves data using various query statements and provides a dedicated user interface for people to access actual graphs. We use two smartwatches, two smart IoT getaways, and a doctor's device to project the overall scenario in Neo4j. We use the Flask module in python and design python programs to project the scenario. The Flask module is used because it supports RESTful API's quick building and is a viable choice for resource-constrained sensor networks [49]. Besides, we deploy a real SSL certificate for the cloud API using Let's encrypt [50]. All the communications with the cloud run on HTTPS.

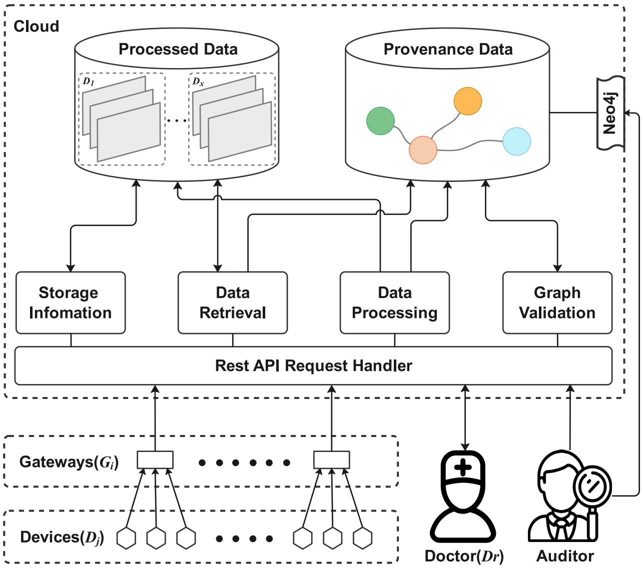


FIGURE 7. Implementation model.

B. EXPERIMENTS

According to the threat model presented in section III-B1, we simulate various security vulnerabilities within this IoT system. We design and simulate attacks to misuse some vulnerabilities and observe whether the scheme is capable enough to detect and locate weak security settings or suspicious changes. We conduct experiments simulating six cases in the following.

Case 1. This case simulates the node cloning attack and fault packets injection attack. An adversary places a malicious node in the system. It fetches information from Tom's smartwatch and copies everything but the secret key of that device. In particular, if the secret key is not used to authenticate the data or the authentication is not validated by the gateway, any data (trusted or untrusted) sent from that node can pass through the IoT gateway. In addition, without sufficient cryptographic protection, the adversary may directly forge or alter packets of Tom's smartwatch and send to the IoT gateway.

Case 2. This case simulates the DoS attack. In this case, we assume an adversary's goal is to exhaust Tom's smartwatch, leading it to go into sleep mode and then generate other possible malicious actions. The attacker sends large volumes of data floods over a period of time to accomplish this attack.

Case 3. An unauthorized access attack is simulated in this case. We assume the victim of this attack is the data generated by Jim's smartwatch. An adversary may exploit attack vectors on the cloud API and force unauthorized access to the 'data retrieve function.' The software version running on the cloud server is an example of relevant security metadata for this case. This is because older versions may have known vulnerabilities with the access control configuration enforced and used by the API, such as requiring multi-factor authentication. Further, log data can reveal attempts for unauthorized access, and large numbers can point to current threats and increased risks.

Case 4. This case simulates another unauthorized access attack on IoT gateway. An adversary may exploit open ports to gain unauthorized access to the IoT gateway. We assume this attack happens when Jim's smartwatch sends data to the cloud, and the gateway can detect the intrusion and give alerts. The changing state of the IoT gateway indicates potential leakage of device management information, which further manipulates the system's privacy.

Case 5. This case simulates a misconfiguration on the cloud server-side, such as not running a web application firewall (WAF). This lack of expected security controls can degrade the trust level. It is worth mentioning that this does not contradict the assumption of a trusted cloud server, as a misconfiguration does not imply any malicious activity from the service provider's side.

Case 6. This case simulates an increased risk of attacks on the health application, e.g., by malicious code injection. The application is running with higher privileges than is required, which increases the risk. Thus, an injected malicious code in a web application can gain more permissions while the doctor is operating/using the application. This malicious code may monitor the doctor's actions, modify the processed data he retrieves or impersonate users.

C. RESULTS AND DETECTION PRINCIPLES

We develop security-aware provenance graphs based on the structure created in IV-A and verify whether the information recorded in security metadata nodes is in consonance with pre-defined security policies presented in IV-B1 after conducting designed attacks. Fig. 8 - 13 reveal the results of six experiments (diagrammatic sketch). Each figure shows information obtained by the doctor and the auditor, respectively. The doctor retrieves the processed data and gets any problematic data name (if any) along with the specific dubious activity of it by evoking the validation function. If any erroneous activity is identified, the doctor may consult the auditor. The auditor then accesses the actual security-aware provenance graph of that problematic file via the Neo4j UI and checks the detailed security metadata of the activity. Thus, the output provides better transparency by showing active security controls, protocols used, or system configurations. We further discuss the detection principles to show how end-users can discover increased risks by validating the security-aware provenance graph for the six cases.

Case 1. The IoT gateway verifies the signature of every piece of data sent by smartwatches. Hence, any untrusted data by a fraudulent node can be identified as it fails the signature verification in the gateway. Suppose malicious data is accepted without passing the validation of the signature. In that case, information about verification failure is recorded in the security metadata node that links to the third activity (the attack happens when data propagates to the cloud via the gateway) which violates pre-defined security policies. An example of identifying failed signature verification for this case is added in Figure 8.

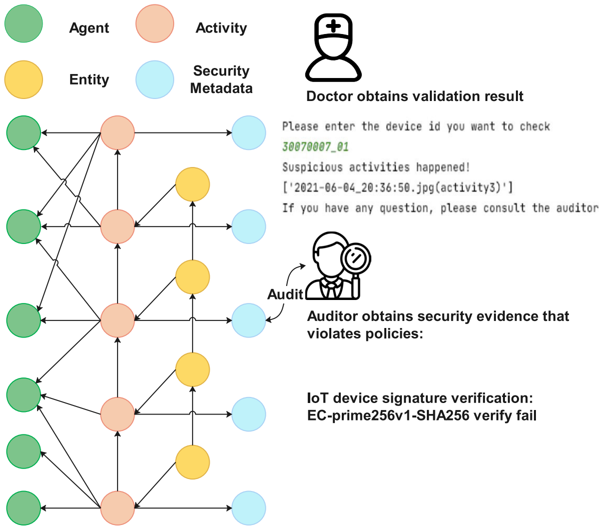
Case 2. An example of detecting DoS with higher power consumption for this case is added in Figure 9. The DoS attack dramatically increases a smartwatch's average power consumption, which exceeds the specified value. The abnormal power consumption data is recorded in the security metadata node that links to the second activity (the attack happens while the sensor is capturing raw data). The validation function gives an alert about the detected anomaly.

Case 3. An example of identifying unauthorized access for this case is added in Figure 10. Records pointing unauthorized access of the same data are stored in each security metadata node. These nodes connect with the fifth activity (it is recorded when adversaries pretend to be the doctor try to retrieve data from the cloud) and reveal possible attacks aimed at the cloud data interface.

Case 4. An example of intrusion detection for this case is added in Figure 11. Intrusion causes the IoT gateway to alert, and this message is recorded in the security metadata node that connects with the third activity (the attack happens when data propagates to the cloud via the gateway). It violates the security policy and the doctor is informed about the security issue.

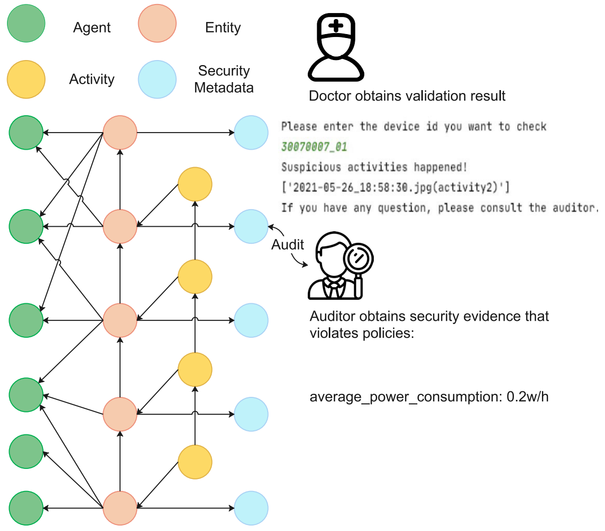
Case 5. An example of identifying missing WAF for this case is added in Figure 12. The corresponding security metadata node records the missing information about the WAF, which is detected during the fifth activity (it is recorded when the doctor retrieves data from the cloud through the web application). As a result, the validation function generates notifications to users to pay attention to potential risks.

Case 6. An example of permission violation for this case is added in Figure 13. According to pre-defined policies, this web application should run with the least privilege to perform all required operations. The evidence that it gains higher permissions is recorded in the security metadata node which links to the fifth activity (it is recorded when the doctor retrieves data from the cloud through the web application).


Agent
Activity
Entity
Security Metadata
Doctor obtains validation result
Audit
Auditor obtains security evidence that violates policies:
IoT device signature verification: EC-prime256v1-SHA256 verify fail

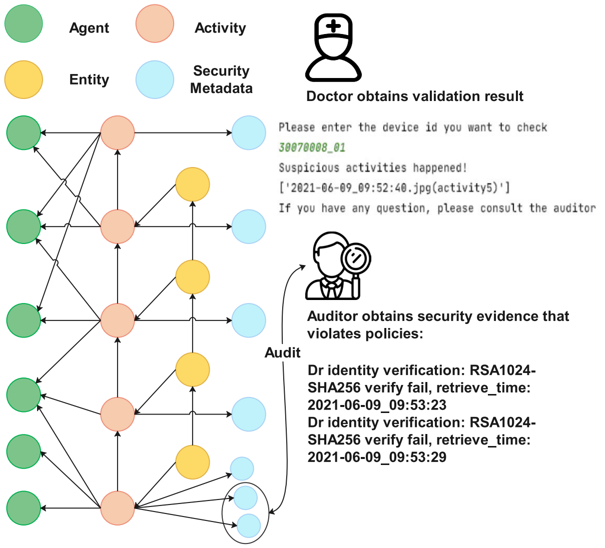
```
Please enter the device id you want to check
30070007_01
Suspicious activities happened!
['2021-06-04_20:36:50.jpg(activity3)']
If you have any question, please consult the auditor
```

FIGURE 8. Experimental result of Case 1.


Agent
Entity
Activity
Security Metadata
Doctor obtains validation result
Audit
Auditor obtains security evidence that violates policies:
average_power_consumption: 0.2w/h

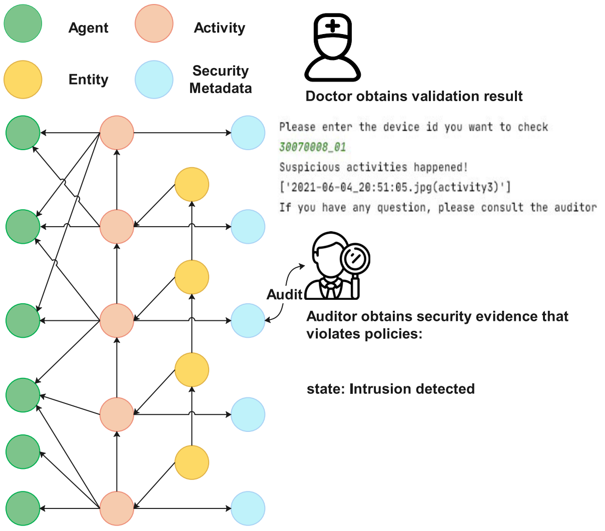
```
Please enter the device id you want to check
30070007_01
Suspicious activities happened!
['2021-05-26_18:58:30.jpg(activity2)']
If you have any question, please consult the auditor.
```

FIGURE 9. Experimental result of Case 2.


Agent
Activity
Entity
Security Metadata
Doctor obtains validation result
Audit
Auditor obtains security evidence that violates policies:
Dr identity verification: RSA1024-SHA256 verify fail, retrieve_time: 2021-06-09_09:53:23
Dr identity verification: RSA1024-SHA256 verify fail, retrieve_time: 2021-06-09_09:53:29

```
Please enter the device id you want to check
30070008_01
Suspicious activities happened!
['2021-06-09_09:52:40.jpg(activity5)']
If you have any question, please consult the auditor
```

FIGURE 10. Experimental result of Case 3.


Agent
Activity
Entity
Security Metadata
Doctor obtains validation result
Audit
Auditor obtains security evidence that violates policies:
state: Intrusion detected

```
Please enter the device id you want to check
30070008_01
Suspicious activities happened!
['2021-06-04_20:51:05.jpg(activity3)']
If you have any question, please consult the auditor
```

FIGURE 11. Experimental result of Case 4.

Consequently, the doctor is notified of the possibility of increased risks.

VI. EVALUATION

Within the IoT scenario, security-aware provenance graphs make the IoT system transparent to end-users. We evaluate the approach from two aspects: a) in obtaining transparency and b) in describing performance analysis.

A. TRANSPARENCY

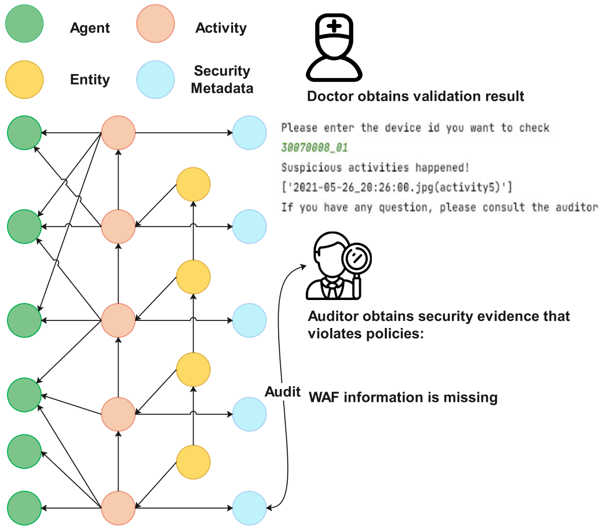
According to the results, we divide security-related problems of six cases into four types: authentication, integrity, availability, and underlying threats. They all contribute to achieving comprehensive transparency, as illustrated below.

a: AUTHENTICATION

End-users in the presented IoT scenario can ensure that medical data or retrieve requests are sent by the correct entities, ensuring system authentication. For the node cloning attack in case 1, the identity of different IoT devices can be recognized by verifying the signatures of those devices. This security evidence is preserved in the security metadata node of activity 3. Hence, it helps end-users to identify (be aware of) which device is forged. Similarly, for an unauthorized access attack in case 3, the cloud server verifies the data retriever's identity. End-users can determine whether data is being received illegally by outsiders by retrieving information saved in the security metadata nodes of activity 5. For an unauthorized access attack in case 4, users other than the authorized technician intruding into the IoT gateway trigger alerts. Hence, a warning message is shown in the security metadata nodes of activity 3.

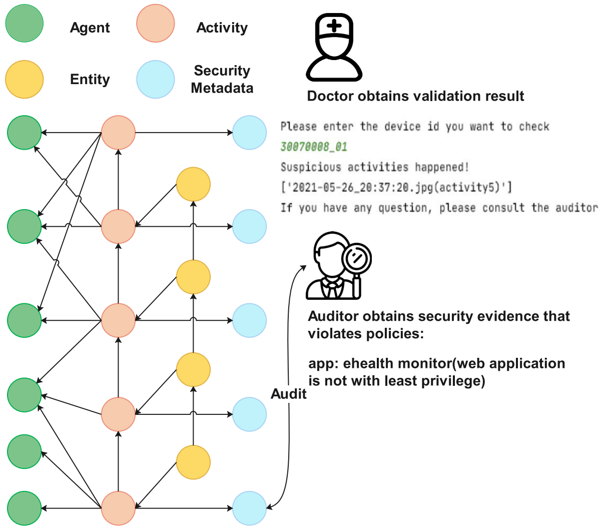
b: INTEGRITY

When the medical data is finally presented on the end-user's screen, integrity should confirm that it accurately represents


Agent
Activity
Entity
Security Metadata
Doctor obtains validation result
Auditor obtains security evidence that violates policies:
Audit
WAF information is missing

```
Please enter the device id you want to check
30070008_01
Suspicious activities happened!
['2021-05-26_20:26:00.jpg(activity5)']
If you have any question, please consult the auditor
```

FIGURE 12. Experimental result of Case 5.


Agent
Activity
Entity
Security Metadata
Doctor obtains validation result
Auditor obtains security evidence that violates policies:
app: ehealth monitor(web application is not with least privilege)
Audit

```
Please enter the device id you want to check
30070008_01
Suspicious activities happened!
['2021-05-26_20:37:20.jpg(activity5)']
If you have any question, please consult the auditor
```

FIGURE 13. Experimental result of Case 6.

the data first captured (the raw data). For example, in case 1, signature verification detects any false data injection in activity 3. Therefore, if the medical data is altered, this security proof precludes the doctor from trusting the data he sees.

c: AVAILABILITY

When any adversary launches attacks to manipulate the availability of the network, our operation scheme can make this action transparent to end-users. In case 2, if any IoT device suffers from a DoS attack, abnormal changes in power consumption are recorded in the security-aware provenance graph. It represents the occurrence of the attack in activity 2 during device operation.

d: UNDERLYING THREATS

Sometimes no substantial attack occurs, or the attack may not cause real damage, but potential threats exist in the system. In case 5, security evidence about incorrect configurations of WAF reveals that this system leaves space for possible attacks in the application layer (activity 5). Additionally, the adversary may launch attacks from end-user's computer instead of IoT gadgets. In case 6, the attacker may steal medical data from doctor's computer directly. However, illegal features of the device software (triggered by the attack) are recorded on particular security metadata nodes of activity 5.

B. PERFORMANCE ANALYSIS

Data propagation in the hypothetical scenario is a continuous process. Multiple graphs are needed to portray the overall scenario. Also, large-scale IoT applications might process data from a very high number of devices. This triggers a significant concern about handling a large number of graphs and raises issues for performance analysis. It questions the applicability of security-aware provenance graphs for big-data IoT applications. However, many applications, such as the scenario depicted in this paper, require data related to individual people or specific objects. In these cases, the complexity mainly depends on the size of individual provenance graphs. Hence, the following paragraphs discuss the estimation of the size of security-aware provenance graphs.

1) UPPER BOUND OF THE SIZE OF THE GRAPH

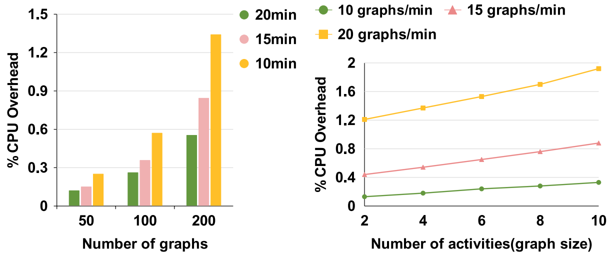
The overhead generated by creating, communicating and processing provenance graphs mainly depends on the growth of the number of nodes in a graph and the need to spawn new graphs for additional devices. The size of the graph depends on the agents, activities, entities, security metadata involved in the provenance graph, and relationships between them. Hence, the growth of provenance graphs at each layer is estimated by nodes and edges we add at each layer of data propagation. It is to be noted that in our envisaged use cases, provenance graphs are built for individual patients and are not intended to be scaled to big data applications. Therefore, it is linear to the size of the graph.

This estimation relies on various parameters on which the size of the graph grows. For example, effort of different policies, and validation among them. Therefore, the growth of the number of nodes in a provenance graph can vary for specific IoT applications.

Considering a patient's m devices (sensors/actors) collect k measurements within time t, N_{PG} is the number of nodes in a provenance graph and N is the number of patients. We calculate the upper bound for the system's number of nodes in a provenance graph as follows.

In the sensing layer, size of the graph (N_{PG}) with m devices of k measurements is represented as O($k \times m$). As it propagates to next layers, the growth of the graph adds step-wise information, not inheriting connections with all other nodes. Therefore, the graph at different layers add the step-wise value with a constant number of new edges and does not grow exponentially. Thus, the size of the graph for one patient after data propagation to application layer would be O($k \times m$).

For the overall system with N patients, separate graphs are generated for each patient. Therefore, the upper bound for the



(a) varying frequency of graph generation (b) varying the graph size

FIGURE 14. Comparison of CPU overhead.

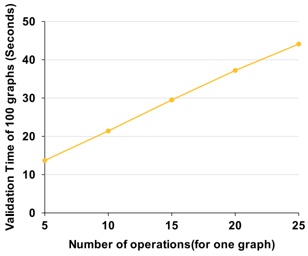


FIGURE 15. Validation time.

total number of nodes and edges to be stored, transmitted and processed is O($k \times m \times N$), which represents a linear growth.

2) CPU OVERHEAD

To measure the CPU overhead caused by graphs, we use a testing tool named Sysbench to acquire CPU performance [51].

First, we identified the CPU performance of ECS (used in section V) when it was running but not generating and storing graphs. Afterward, we configured the program in ECS to generate provenance graphs at different frequencies and recorded CPU performance (the number of activities in a graph is 4). Finally, we varied the number of activities in a graph (graph size) and calculated the CPU overhead. During the test, we turned off unnecessary functions to ensure extra CPU overhead is from generating and storing provenance graphs.

Fig. 14a illustrates that higher frequency of graph processing adds CPU overhead dramatically. It indicates more IoT sensors in real-life scenarios significantly burden the CPU and consume higher CPU power. Processing 200 graphs within 10 minutes caused highest CPU overhead in the test, which was 1.34%. However, the required computational power was in the expected range. We notice from Fig. 14b that CPU overhead grows slightly with the increment of graph size and higher graph processing frequency results higher rate of growth in CPU overhead. The performance is adequate for handling large-sized graphs with more content. It would be more adaptive in complex IoT environments that produce larger graphs.

3) VALIDATION TIME

Validation time is another feature closely linked with performance. We recorded validation time of 100 graphs. We varied the number of security policies used by validation function to adjust the required number of operations for validating each graph. Based on the information in Fig. 15, with the increase of required operations for each graph, time consumption of validating a 100 graphs rises modestly. For example, validation time increases by only 17.4% as the number of operations for each graph varies from 20 to 25. Considering system transparency is vital in certain IoT scenarios, this level of time consumption is within an acceptable range.

Apart from effecting validation time, the number of security policies that the validation function uses also indicates the transparency it offers. In practice, it is undesirable to use insufficient or redundant security policies. While redundancy can improve system transparency, if it simply indicates an overlap in what is validated, it does not promote transparency and causes longer validation times. Conversely, insufficient policies reduce the transparency level and increase risk. To sum up, security policies should be updated (when necessary) to keep a balance between system transparency and validation time.

In this paper, we developed and evaluated a first proof-of-concept implementation of the framework of security-aware provenance from an end-user perspective. The validation process covers multiple graphs representing different data sources or objects in an IoT environment. The evaluation shows that it is, in principle, practical to specify security policies that enable automatic validation and provide detailed results to the end-user. The end-user sees an abstract result of the validation, and if there is any doubt, he can interactively explore the details of the stored provenance graphs. Meanwhile, graph establishment is synchronized with data transmission and the steps in the process. Graphs can be automatically spawned for new data sources.

Mapping graphs to instances in the application is based on identifiers for each graph, including information about the registered data generator (IoT device), the registered user (i.e., patient), and when aggregated data is processed at the cloud. Thus, the validation function can distinguish multiple security-aware provenance graphs for different patients, IoT devices and separate transmissions in accordance with identifiers. Accurate mapping and automatic validation of graphs are critical to scaling to more significant numbers of devices and users.

With this regard, it is worth mentioning that the proposed security-aware provenance graph implementation is designed for IoT-health applications, where a particular patient data is a concern, and we focus on a particular number of devices. In practice, IoT applications with similar setups need to audit a particular number of devices with specific instance(s) for identifying transparency or risk estimation. Hence, in the end, it generates a linear growth in graphs in terms of handling the devices. Many practical IoT applications fall into this class of processes with linear growth of graph numbers depending on sensors and data flows. For all these applications, the suggested approach of security-aware provenance graphs can therefore be assumed to be feasible and scalable to a higher number of users providing the following properties.

- Data sources need to be authentic.
- Changes to data of a single sensor or a small set of sensors can cause substantial damage.
- The relation between sensors and humans or objects is essential.
- Risk mitigation cannot easily rely on redundancy achieved by a large number of sensors or by statistical corrections to the data.

VII. IMPLICATIONS AND LIMITATIONS

The proposed scheme can be applied in many smart environments. For example, services in smart homes in monitoring water meters in an apartment building. While it can increase trust into the metering for the end user, it can also be convenient for the apartment administrator to conduct forensic tasks if any part of the smart water meters in an apartment is manipulated by an adversary or in an unusual condition. For a large-scale smart city project, the implementation would need to be optimized with more elastic provenance graph management and more robust data processing ability. However, the current implementation demonstrates its capability of supporting these kinds of applications.

The current version of security-aware provenance graphs is not directly suitable for big data IoT applications where thousands of nodes collect data in a data lake and then used by various applications. However, even in such a scenario, it would be possible to add similarly enhanced provenance information to every single piece of data in the data lake. Therefore, we expect that it is possible to transfer the approach to these large-scale scenarios. Secondly, the automated validation function is deployed in the cloud, and we assume the cloud is trusted in this scenario. One potential risk is that if the cloud encounters internal problems or there is an insider attack on the cloud provider's side, the validation function may become unavailable, or the results can be misleading. Finally, it is probably challenging to develop generic policy templates. Each adaptive IoT scenario has different requirements and requires dedicated security policies, which raises this scheme's deployment cost. However, these costs might still be smaller than the costs of successful attacks.

VIII. CONCLUSION AND FUTURE WORK

The challenge of how to maintain system transparency in IoT scenarios is gradually emerging. Data provenance graphs with security metadata are considered a potential solution for this kind of issue. This paper presents the trustworthiness in IoT data propagation by comprehensive transparency. To acquire security-aware features, we first describe a hypothetical IoT-health scenario and generate a relevant provenance graph that includes security metadata as well as its relationships on standard provenance graphs. The experimental results confirm that the upgraded provenance graph can help end-users make appropriate decisions by monitoring and locating the problematic parts. Simultaneously, it can provide precise security proof while also providing professionals with new insight into potential threats and points to active attacks.

In the future, we plan to explore and incorporate the security awareness feature of provenance graphs into intelligent automated visualization. During the experiment process, we become aware that provenance graphs are flexible enough to serve additional purposes. For instance, entity nodes of provenance graphs represent real data in an IoT system. Therefore, the provenance graph can serve two functions simultaneously. One is for security evaluation. Another one is to act as a platform to present data to the end-user, especially when the data needs to be processed into graphs. Hence, apart from security-aware ability, what else functions can we develop to serve IoT systems, even advanced data-driven AI applications, can be additional research questions.

Moreover, validation of the security metadata used at each step of data propagation is essential and needs additional research. In practical, the security metadata used at each activity of provenance graphs need to be validated step-by-step during the process. Methods on efficient handling of this step-wise validation and final evaluation to be addressed by end-users based on an accessible visualization of the graph are open research issues.

REFERENCES

[1] S. Balakrishnan, H. Vasudavan, and R. K. Murugesan, "Smart home technologies: A preliminary review," in *Proc. 6th Int. Conf. Inf. Technol., IoT Smart City (ICIT)*. New York, NY, USA: Association for Computing Machinery, Dec. 2018, pp. 120–127, doi: 10.1145/3301551.3301575.

[2] R. De Michele and M. Furini, "IoT healthcare: Benefits, issues and challenges," in *Proc. 5th EAI Int. Conf. Smart Objects Technol. Social Good (GoodTechs)*. New York, NY, USA: Association for Computing Machinery, Sep. 2019, pp. 160–164, doi: 10.1145/3342428.3342693.

[3] R. Lea and M. Blackstock, "Smart cities: An IoT-centric approach," in *Proc. Int. Workshop Web Intell. Smart Sens. (IWWISS)*. New York, NY, USA: Association for Computing Machinery, Sep. 2014, pp. 1–2, doi: 10.1145/2637064.2637096.

[4] A. Harit, A. Ezzati, and R. Elharti, "Internet of Things security: Challenges and perspectives," in *Proc. 2nd Int. Conf. Internet Things, Data Cloud Comput. (ICC)*. New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 1–8, doi: 10.1145/3018896.3056784.

[5] A. Q. Gill, V. Behbood, R. Ramadan-Jradi, and G. Beydoun, "IoT architectural concerns: A systematic review," in *Proc. 2nd Int. Conf. Internet Things, Data Cloud Comput. (ICC)*. New York, NY, USA: Association for Computing Machinery, Mar. 2017, pp. 1–9, doi: 10.1145/3018896.3025166.

[6] Y. H. Hwang, "IoT security & privacy: Threats and challenges," in *Proc. 1st ACM Workshop IoT Privacy, Trust, Secur. (IoTPTS)*. New York, NY, USA: Association for Computing Machinery, 2015, p. 1, doi: 10.1145/2732209.2732216.

[7] P. M. Rao and B. D. Deebak, "Security and privacy issues in smart cities/industries: Technologies, applications, and challenges," *J. Ambient Intell. Humanized Comput.*, pp. 1–37, Feb. 2022.

[8] W. Zhang, I. A. Elgendy, M. Hammad, A. M. Iliyasu, X. Du, M. Guizani, and A. A. A. El-Latif, "Secure and optimized load balancing for multitier IoT and edge-cloud computing systems," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8119–8132, May 2021.

[9] A. A. A. El-Latif, B. Abd-El-Atty, S. E. Venegas-Andraca, H. Elwahsh, Md. J. Piran, A. K. Bashir, O. Song, and W. Mazurczyk, "Providing end-to-end security using quantum walks in IoT networks," *IEEE Access*, vol. 8, pp. 92687–92696, 2020.

[10] F. T. Jaigirdar, C. Rudolph, and C. Bain, "Can I trust the data I see? A Physician's concern on medical data in IoT health architectures," in *Proc. Australas. Comput. Sci. Week Multiconference*, Sydney, NSW, Australia, Jan. 2019, pp. 1–10.

[11] F. T. Jaigirdar, C. Rudolph, G. Oliver, D. Watts, and C. Bain, "What information is required for explainable AI?: A provenance-based research agenda and future challenges," in *Proc. IEEE 6th Int. Conf. Collaboration Internet Comput. (CIC)*, Dec. 2020, pp. 177–183.

[12] F. T. Jaigirdar, C. Rudolph, and C. Bain, "Risk and compliance in IoT-health data propagation: A security-aware provenance based approach," in *Proc. IEEE Int. Conf. Digit. Health (ICDH)*, Sep. 2021, pp. 27–37.

[13] M. Bada, A. M. Sasse, and J. R. C. Nurse, "Cyber security awareness campaigns: Why do they fail to change behaviour?" 2019, *arXiv:1901.02672*.

[14] T. T. Dayaratne, F. T. Jaigirdar, R. Dasgupta, A. Sakzad, and C. Rudolph, "Improving cybersecurity situational awareness in smart grid environments," in *Power Systems Cybersecurity: Methods, Concepts, and Best Practices*. Springer, 2023, pp. 115–134.

[15] Z.-K. Zhang, M. C. Y. Cho, and S. Shieh, "Emerging security threats and countermeasures in IoT," in *Proc. 10th ACM Symp. Inf., Comput. Commun. Secur. (ASIA CCS)*. New York, NY, USA: Association for Computing Machinery, Apr. 2015, pp. 1–6, doi: 10.1145/2714576.2737091.

[16] F. T. Jaigirdar, C. Rudolph, and C. Bain, "Prov-IoT: A security-aware IoT provenance model," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1360–1367.

[17] M. N. Aman, K. C. Chua, and B. Sikdar, "Secure data provenance for the Internet of Things," in *Proc. 3rd ACM Int. Workshop IoT Privacy, Trust, Secur.*, Apr. 2017, pp. 11–14.

[18] M. H. Chia, S. L. Keoh, and Z. Tang, "Secure data provenance in home energy monitoring networks," in *Proc. 3rd Annu. Ind. Control Syst. Secur. Workshop (ICSS)*. New York, NY, USA: Association for Computing Machinery, Dec. 2017, pp. 7–14, doi: 10.1145/3174776.3174778.

[19] A. Alkhalil and R. A. Ramadan, "IoT data provenance implementation challenges," *Proc. Comput. Sci.*, vol. 109, pp. 1134–1139, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917311183

[20] R. Hasan, R. Sion, and M. Winslett, "Introducing secure provenance: Problems and challenges," in *Proc. ACM Workshop Storage Secur. Survivability (StorageSS)*. New York, NY, USA: Association for Computing Machinery, 2007, pp. 13–18, doi: 10.1145/1314313.1314318.

[21] U. Braun, A. Shinnar, and M. Seltzer, "Securing provenance," in *Proc. 3rd Conf. Hot topics Secur. (HOTSEC)*, 2008, pp. 1–5.

[22] O. I. Abiodun, M. Alawida, A. E. Omolara, and A. Alabdulatif, "Data provenance for cloud forensic investigations, security, challenges, solutions and future perspectives: A survey," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 10217–10245, Nov. 2022.

[23] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, and J. V. den Bussche, "The open provenance model core specification (v1.1)," *Future Gener. Comput. Syst.*, vol. 27, no. 6, pp. 743–756, Jun. 2011, doi: 10.1016/j.future.2010.07.005.

[24] R. Hasan, R. Sion, and M. Winslett, "Preventing history forgery with secure provenance," *ACM Trans. Storage*, vol. 5, no. 4, pp. 1–43, Dec. 2009.

[25] N. Baracaldo, L. A. D. Bathen, R. O. Ozugha, R. Engel, S. Tata, and H. Ludwig, "Securing data provenance in Internet of Things (IoT) systems," in *Proc. Int. Conf. Service-Oriented Comput.* Springer, 2016, pp. 92–98.

[26] J. Li, X. Chen, Q. Huang, and D. S. Wong, "Digital provenance: Enabling secure data forensics in cloud computing," *Future Gener. Comput. Syst.*, vol. 37, pp. 259–266, Jul. 2014.

[27] S. Zawoad, R. Hasan, and K. Islam, "SECProv: Trustworthy and efficient provenance management in the cloud," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 1241–1249.

[28] S. Khatal, J. Rane, D. Patel, P. Patel, and Y. Busnel, "FileShare: A blockchain and IPFS framework for secure file sharing and data provenance," in *Advances in Machine Learning and Computational Intelligence*. Springer, 2021, pp. 825–833.

[29] P. Missier, J. Bryans, C. Gamble, and V. Curcin, "Abstracting PROV provenance graphs: A validity-preserving approach," *Future Gener. Comput. Syst.*, vol. 111, pp. 352–367, Oct. 2020.

[30] M. Sigwart, M. Borkowski, M. Peise, S. Schulte, and S. Tai, "Blockchain-based data provenance for the Internet of Things," in *Proc. 9th Int. Conf. Internet Things*, Oct. 2019, pp. 1–8.

[31] D. Batista, H. Kim, V. L. Lemieux, H. Stancic, and C. Unnithan, "Blockchains and provenance: How a technical system for tracing origins, ownership and authenticity can transform social trust," in *Building Decentralized Trust*. Springer, pp. 111–128.

[32] M. Frustaci, P. Pace, G. Aloi, and G. Fortino, "Evaluating critical security issues of the IoT world: Present and future challenges," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2483–2495, Aug. 2018.

[33] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017.

[34] S. Valluripally, A. Gulhane, R. Mitra, K. A. Hoque, and P. Calyam, "Attack trees for security and privacy in social virtual reality learning environments," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2020, pp. 1–9.

[35] Y. Gu, P. Zhang, Z. Chen, and F. Cao, "UEFI trusted computing vulnerability analysis based on state transition graph," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2020, pp. 1043–1052.

[36] R. Sahay, G. Geethakumari, and K. Modugu, "Attack graph—Based vulnerability assessment of rank property in RPL-6LOWPAN in IoT," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 308–313.

[37] E. Pignotti and P. Edwards, "Trusted tiny things: Making the Internet of Things more transparent to users," in *Proc. Int. Workshop Adapt. Secur.*, Sep. 2013, pp. 1–4.

[38] M. Markovic, D. Garijo, P. Edwards, and W. Vasconcelos, "Semantic modelling of plans and execution traces for enhancing transparency of IoT systems," in *Proc. 6th Int. Conf. Internet Things, Syst., Manage. Secur. (IOTSMS)*, Oct. 2019, pp. 110–115.

[39] E. Nwafor, A. Campbell, D. Hill, and G. Bloom, "Towards a provenance collection framework for Internet of Things devices," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2017, pp. 1–6.

[40] R. K. Lomotey, J. C. Pry, and C. Chai, "Traceability and visual analytics for the Internet-of-Things (IoT) architecture," *World Wide Web*, vol. 21, no. 1, pp. 7–32, Jan. 2018.

[41] J. Singh, J. Cobbe, and C. Norval, "Decision provenance: Harnessing data flow for accountable systems," *IEEE Access*, vol. 7, pp. 6562–6574, 2019.

[42] M. Markovic, D. Corsar, W. Asif, P. Edwards, and M. Rajarajan, "Towards transparency of IoT message brokers," in *Proc. Int. Provenance Annotation Workshop*. Springer, 2018, pp. 200–203.

[43] D. G. Páez, F. Aparicio, M. de Buenaga, and J. R. Ascanio, "Big data and IoT for chronic patients monitoring," in *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, R. Hervás, S. Lee, C. Nugent, and J. Bravo, Eds. Cham, Switzerland: Springer, 2014, pp. 416–423.

[44] A. Mosenia and N. K. Jha, "A comprehensive study of security of Internet-of-Things," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 4, pp. 586–602, Oct. 2017.

[45] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, Jul. 2013, doi: 10.1016/j.comnet.2012.12.018.

[46] M. Bond and R. Anderson, "API-level attacks on embedded systems," *Computer*, vol. 34, no. 10, pp. 67–75, 2001.

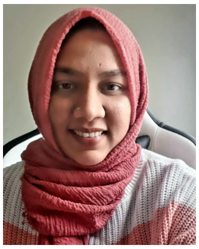
[47] W. Iqbal, H. Abbas, M. Daneshmand, B. Rauf, and Y. A. Bangash, "An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10250–10276, Oct. 2020.

[48] J. Saleem, M. Hammoudeh, U. Raza, B. Adebisi, and R. Ande, "IoT standardisation: Challenges, perspectives and solution," in *Proc. 2nd Int. Conf. Future Netw. Distrib. Syst. (ICFNDS)*. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–9, doi: 10.1145/3231053.3231103.

[49] G. Mainland, G. Morrisett, and M. Welsh, "Flask: Staged functional programming for sensor networks," in *Proc. 13th ACM SIGPLAN Int. Conf. Funct. Program.*, Sep. 2008, pp. 335–346.

[50] J. Aas, R. Barnes, B. Case, Z. Durumeric, P. Eckersley, A. Flores-López, J. A. Halderman, J. Hoffman-Andrews, J. Kasten, E. Rescorla, S. Schoen, and B. Warren, "Let's encrypt: An automated certificate authority to encrypt the entire web," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 2473–2487, doi: 10.1145/3319535.3363192.

[51] A. Kopytov. (2004). *Sysbench: A System Performance Benchmark*. [Online]. Available: http://sysbench.sourceforge.net/



FARIHA TASMIN JAIGIRDAR (Member, IEEE) received the Ph.D. degree in cybersecurity from the Faculty of Information Technology, Monash University, Australia, in 2021. She is currently a Postdoctoral Researcher with the Department of Software Systems and Cybersecurity, Faculty of Information Technology, Monash University. She has been in security-related research for more than nine years. She has published several articles in renowned conferences and journals. Her research interests include different security applications of the IoT, system engineering design and security analysis, trust and risk investigation, and security in health informatics. She is a member of ACM, Australian Women in Security Network, and N2Women Network. She was a recipient of the 2021 Women in Service Computing, and the 2022 Dean's Award for Equity, Diversity and Inclusion from the Faculty of IT, Monash University.

BOYU TAN is currently a Cyber Security Consultant with China Mobile Group Design Institute Company Ltd. His research interests include cybersecurity and security validation.

CARSTEN RUDOLPH is currently the Deputy Dean and a Professor of cybersecurity with the Faculty of Information Technology, Monash University, and the Research Director of the Oceania Cyber Security Centre (OCSC), Melbourne, Australia, where he collaborates with Oxford University to carry out cybersecurity maturity reviews with nations in the Pacific region. He contributes to the development of secure solutions for digital health as well as future energy networks. Further, he drives scientific exchange between cybersecurity, law, and organizational informatics. His research interests include information security, formal methods, cryptographic protocols, security of machine learning, and human aspects of security with a strong focus inter-disciplinary topics. Another focus of his research is on nation-level cybersecurity maturity and policy development.

CHRIS BAIN is an Inaugural Professor (Practice) of digital health with the Faculty of Information Technology (IT), Monash University. He is also an Adjunct Professor with the Faculty of Medicine, Nursing and Health Sciences, Monash University. As an Adjunct Professor, he leads the Digital Health Theme with the Faculty of IT, Monash University. His background is a 30 year history in healthcare and health informatics in Australia, most of that time in serving in operational and operational leadership roles. He is one of the very few medical practitioners who is also fully qualified as an IT professional with both the ACS, Australia, and the ACM, USA. His personal career interests reside in the areas of the use of technology and data for health delivery improvement in all healthcare settings, with a particular interest in usability and outcomes. He has a unique skill set and experience with which to follow through on this passion. He continues to be a thought leader in this area, with a national profile in the health informatics (digital health) domain in Australia.