SETUCOM: Secure and Trustworthy Context Management for Context-Aware Security and Privacy in the Internet of Things

Tidiane Sylla, Mohamed Aymen Chalouf, Francine Krief, Karim Samaké

Hindawi
Security and Communication Networks
Volume 2021, Article ID 6632747, 24 pages
https://doi.org/10.1155/2021/6632747

Research Article

SETUCOM: Secure and Trustworthy Context Management for Context-Aware Security and Privacy in the Internet of Things

Tidiane Sylla,[1,2] **Mohamed Aymen Chalouf**,[3] **Francine Krief**,[1] **and Karim Samaké**[2]

[1]*University of Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR 5800, Talence 33400, France*
[2]*University of Sciences Techniques and Technologies of Bamako, FST-ISA, Bamako, Mali*
[3]*University of Rennes 1, CNRS, IRISA Lab, UMR 6074, Lannion 22300, France*

Correspondence should be addressed to Tidiane Sylla; tidiane.sylla@u-bordeaux.fr

Received 2 January 2021; Revised 16 March 2021; Accepted 25 March 2021; Published 14 April 2021

Academic Editor: Mamoun Alazab

IoT technologies facilitate the development and the improvement of pervasive computing by enabling effective context-awareness features. These features enable the IoT applications to detect the user's situation and adapt their behavior. They also enable context-aware security and privacy, which consist in adapting security and privacy mechanisms' deployment to the user's situation. Research studies on context-aware security and privacy focus on security and privacy mechanisms' implementation but do not consider the secure and trustworthy context management. In this paper, we introduce a new secure and trustworthy context management system for context-aware security and privacy in the smart city: "SETUCOM." SETUCOM is the implementation of the DTM (Device Trust Management) module of the CASPaaS (Context-Aware Security and Privacy as a Service) architecture. It secures context information exchange by using a lightweight hybrid encryption system adapted to IoT devices and manages trust through artificial intelligence techniques such as Bayesian networks and fuzzy logic. A detailed description of the proposed system is provided, and its main performances are evaluated. The results prove SETUCOM feasibility in context-aware security and privacy for the smart city.

1. Introduction

The smart city's applications expand the boundaries of conventional computing by interconnecting the physical world, embedded devices, sensors, or any other electronic device. They enable the collection of information about the physical world. Based on this information, changes are detected, and adaptations are made according to new conditions. Indeed, the data collected by the sensors are analyzed (usually in the cloud) using complex algorithms, and decisions are made. However, the implementation of these applications involves several security and privacy issues [1]. Several research studies have led to the development of security solutions. One solution is the implementation of context-aware security [2, 3]. Indeed, given the frequent changes that characterize IoT, it may be relevant to adapt the security and privacy mechanisms to the current context.

A context can be defined as a set of information that makes possible to determine the situation of an entity (e.g., person). The information used to determine a context is called context information. For example, we have GPS position, presence of a RFID tag, motion speed, and gyroscope value. Context-aware security and privacy in IoT consist in using relevant context information provided by IoT devices to adapt security and privacy mechanisms to the actual context, without explicit user intervention [4]. This could be performed by defining and implementing security and privacy mechanisms adapted to each user's context (situation). Indeed, in IoT, the user's context could change frequently, and each situation has its own risks and threats. Context-aware security can take into account the growing and different threats within IoT. For example, instead of static three-factor authentication as proposed in [5], the Context Adaptive Authentication Service (CAAS) allows a user to dynamically choose a type of authentication: simple

(one-factor) or strong (two-factor or three-factor). The chosen authentication method depends on the risk associated to the user's context [6]. Context-aware authorization management allows the user to grant or deny access to the controlled resources based on the context and the rights of an entity. Thus, a context-aware security and privacy architecture supporting the most relevant security and privacy services has been proposed in [7].

Figure 1 illustrates a context-aware security and privacy system for smart city applications. User's devices transmit context information to the context manager. The context manager models and merges this information to identify the context (approaching home, in the office, etc.). The context security manager then uses the identified context to select and deploy security and privacy mechanisms corresponding to this context. For example, the system will ask the user to perform a strong authentication (PIN code and fingerprint).

However, a context-aware security and privacy environment faces many security and trust issues. First, it is possible for an adversary to monitor such a system, intercept context information, and change it in order to mislead the perception of the system. Indeed, context-awareness management systems usually use unsecure context information [8]. Second, adversaries may fraudulently introduce malicious devices into these systems. Third, legitimate user devices may have faulty sensors. Thus, context-awareness management systems could perform dynamic adaptations based on false context information. For example, an attacker could aim at distorting the deployment of security mechanisms. Then, this attacker can have access to the user's home control applications without authentication and thus enter the house. So, securing the exchange of context information in context-aware systems is very important. In this sense, Zuo et al. [9] investigated and evaluated information security issues in IoT. They proposed a unified framework for information security evaluation in the IoT systems. Chen et al. [10] proposed a new technique to guarantee secure access control, which preserves data authenticity and integrity in IoT. Le Nguyen et al. [11] proposed a blockchain-based technique for secure and reliable IoT data sharing.

Trust management is critical in a context-aware security and privacy environment. It helps to detect devices that behave maliciously. Hence, it enables preventing attacks such as device cloning, spoofing, and context information forging [12]. The thorough and in-depth verification of the reliability of context information allows the context-awareness management system to avoid dealing with inaccurate and/or malicious context information. So, the system can make reliable adaptation decisions using secure and trustworthy context information from trusted devices. Therefore, the design and the implementation of secure and trustworthy context-awareness management become a necessity. This system will use reliable and secure context information coming from trusted devices.

That is why this paper proposes a secure and trustworthy context-awareness management system. The proposed system includes a new mechanism for the secure delivery of context information. It also includes a new reputation-based trust management mechanism. Thus, this system ensures the integrity, confidentiality, and protection against replay of context information. It also allows the detection of suspicious and malicious devices as well as bad context information using artificial intelligence techniques. Therefore, our system can mitigate spoofing, tampering, and eavesdropping attacks while preserving privacy. It will also resist to ballot stuffing attacks, context information forging, and spoofing attacks. In addition, it has the advantage of being lightweight and, so, adapted IoT environments. Even if this paper focuses on the smart city and its applications, it will be easy to integrate the proposed system into other IoT systems such as e-Health.

We present the design of the proposed system, and we evaluate its effectiveness and performances. The major contributions of this paper can be summarized as follows:

(1) The proposal of SETUCOM, integrated secure context-awareness management in a trusted IoT environment, security

(2) The proposal of an evaluation scheme for the reliability of context information based on Bayesian networks that combines context information with the user's profile

(3) The proposal of a context source behavior evaluation scheme based on the fuzzy logic that computes context source-related statistics (old and actual states) to determine their behavior (good, doubtful, or malicious)

The rest of this paper is organized as follows. Section 2 compares existing solutions for secure management of context awareness in a trusted environment. Section 3 presents the key features of the CASPaaS (Context-Aware Security and Privacy as a Service) security architecture proposed in [7] on which our solution is based. Section 4 describes the proposed system, and Section 5 evaluates the effectiveness and the main performances of this system. Finally, Section 6 concludes the paper and points out some relevant perspectives.

2. Related Work

Several studies focused on context-aware security and privacy in the smart city. In [8], the authors reviewed the security issues of context-aware systems. Mahalle and Dhotre [13] described the security requirements for a context-aware system. However, several research studies that addressed the deployment of context-aware security and/or privacy mechanisms in IoT did not consider security, trust of context sources, and reliability of context information [2, 6, 14–16]. This section reviews the few works that considered these issues.

2.1. Secure Exchange of Context Information. Secure exchange of context information is very important in context-awareness management. Indeed, adversaries may monitor the system, attempt to replicate contexts, and to mislead system perception. Ahamed et al. [17] addressed the problem

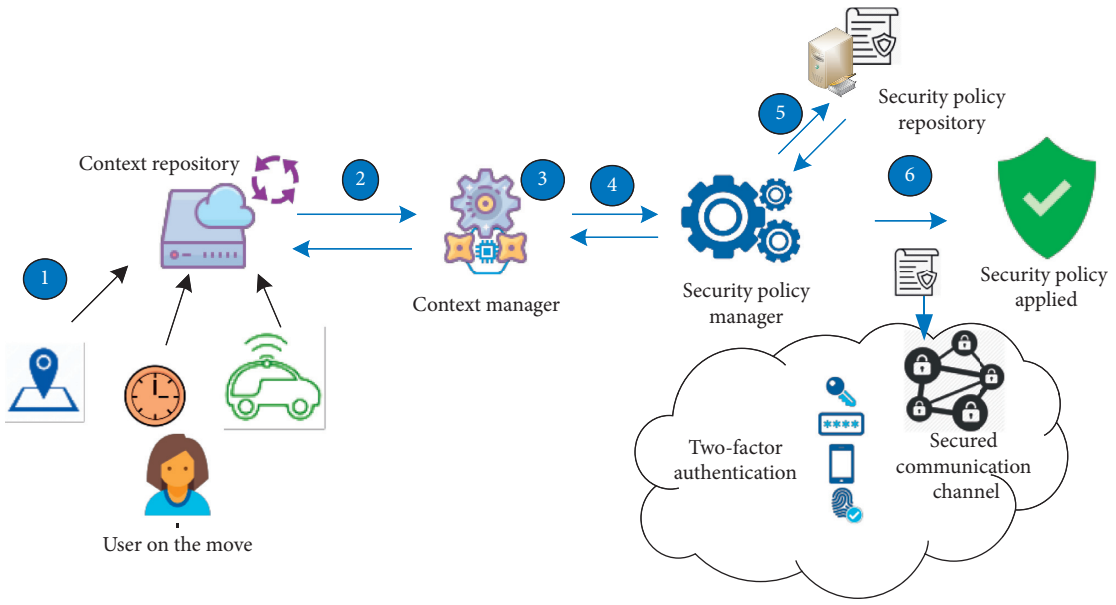


Figure 1: Context-aware security implementation [4].

of context-aware authentication and access control in healthcare IoT systems. They proposed a new mechanism called Enhanced Context-aware Capability-based Access Control (ECCAPAC). The proposed mechanism enforces access control by using a capability tag and context information. However, the security of context information exchange is not considered in this work.

In [18], Alagar et al. proposed a context-aware role-based access control system for a hospital e-Health system. In this solution, the authors propose the implementation of information transmission security. To achieve that, they propose the use of mechanisms provided for this purpose in the IEEE 802.15.6 standard. Indeed, this standard provides the implementation of authenticated and confidential WBAN (Wireless Body Area Network) intrabody network communications. However, the range of a WBAN is limited to the carrier's body. It does not ensure information security for extra-WBAN communications. In addition, the solutions proposed in [3, 18] were neither implemented nor evaluated.

Chouhan et al. [19] highlighted the security issues in IoT environments and proposed to use the concept of situation evaluation to ensure the security of IoT applications. This concept is based on the detection of events that can help to evaluate the situation and propose appropriate security mechanisms. However, the security of the data provision allowing situation evaluation and events' detection has not been addressed. The confidence of the devices providing these data has also not been considered. In addition, the proposed concept has not been evaluated.

Ashibani et al. [6] proposed a context-aware authentication service for smart home applications. The major advantage of the proposed scheme is that it helps to reinforce the user's authentication and access control by making them adaptive to the user's context. However, context information gathering is not secured. de Matos et al. [2] proposed a context-aware security module for edge-centric context sharing architecture. This solution takes advantage from leveraging edge/fog computing infrastructure integration. This enables low latency in transmitting context information and supporting mobility. Nevertheless, the context information transmission security is not ensured in this architecture.

However, securing communications in IoT environments has been the object of several research studies. In OSCAR [20], the authors proposed a mechanism for encrypting CoAP (Constrained Application Protocol) [21] data packets exchanged over an unsecure network. OSCAR addresses the limitations of the Datagram Transport Layer Security (DTLS) protocol by encrypting the application layer payload. In [22], the authors based their work on OSCAR to propose IoTChain. Unlike OSCAR, trust, in IoTChain, is decentralized and managed by the blockchain. This system has the advantage of not having a centralized trust root. It also overcomes the limitations of the DTLS protocol. However, these solutions (OSCAR and IoTChain) involve several systems, including key servers, resource servers (using CoAP), and proxy servers for the blockchain. This makes these solutions cumbersome for information exchange security. As a result, they are not suitable for a context-aware security and privacy system in IoT. In addition, MQTT (Message Queuing Telemetry Transport) protocol is preferred to CoAP in collecting context

information. This is mainly due to the performance of MQTT, which is 30% more efficient than CoAP for the transmission of the same payloads [23].

Malina et al. [24] proposed a security architecture for the MQTT protocol. The proposed solution provides three layers of security for secure communications under MQTT. They implemented and evaluated the solution. The most robust security layer requires a trusted third party. However, a trusted third party adds an overlay that could make the context-aware security and privacy architecture more cumbersome.

2.2. Trust Management. In a context-aware system, the trust management of context sources and the reliability control of context information are crucial. In this sense, Arfaoui et al. [25] considered the problem of authorization management in IoT. They proposed a context-aware access control mechanism named Context-Aware Attribute-Based Access Control (CAABAC). The proposed mechanism uses context information and users' attributes to define access control rules. However, in this work, the authors did not tackle context information reliability and the context sources' trust. As a result, the mechanism can use false context information to make bad access control decision.

Furthermore, several research studies have been performed on trust management in smart city applications. Chen et al. [26] proposed a trusted architecture for IoT, including a cross-layer authorization protocol in a Software-Defined Networking (SDN) network and a reputation management layer. The authors described two mechanisms, respectively, called behavior-based reputation evaluation scheme and organization reputation evaluation scheme, for the evaluation of node and organization behavior. However, these reputation management mechanisms are coupled with permission management. This implies additional overhead for context-aware security and privacy.

Ensuring security and trust of computer networks and IoT environments, in particular, using artificial intelligence techniques is a promising research field. In this context, several research studies had been done for the detection of malicious behaviors of devices. These research studies focused on artificial intelligence techniques such as fuzzy logic, machine and deep learning, and Bayesian networks. In this sense, Swarna Priya et al. [27] proposed an efficient Deep Neural Network-based Intrusion Detection System (IDS) for medical IoT systems. The proposed system detects and classifies attacks efficiently. Rehman et al. [28] proposed DIDDOS, a solution that aims at detecting and mitigating distributed denial-of-service (DDoS) attacks in computer networks. The proposed solution uses Gate Recurrent Unit (GRU). The evaluation proved the solution efficiency in DDoS attacks' detection. For example, Shafiq et al. [29] proposed a new Machine Learning- (ML-) based technique for detecting malicious IoT botnet traffic in a smart city. The proposed technique and its efficiency have been evaluated compared to other techniques in the literature.

In the same context, Rehman Javed et al. [30] proposed a new approach for the detection of botnet attacks in connected vehicle networks. This approach uses machine learning algorithms on network traffic. According to the evaluation results, the proposed scheme has good efficiency compared to other solutions in the same area. Nevertheless, these proposals [27–32] are adapted to IoT environments with high object density like in the smart city. Unfortunately, these proposals aim at protecting IoT environments from well-known and specific networks' attacks. The data used for device reputation assessment are not available in a context-aware security and privacy environment for the smart city. There is no interaction between devices in such an environment. Indeed, exchanges take place between the system and the context sources. Therefore, additional data are required to enable dynamic reputation assessment in this environment (user profile, experiences, period of operations, etc.).

The thorough verification of information reliability is complementary to trust management. The approaches described in [3, 26, 33] do not consider this dimension in trust management. It is very important to consider this aspect because it provides additional leverage to assess the credibility of the context sources themselves, which will allow the system to detect malicious devices.

The above-summarized research studies have proposed solutions for context-aware security in a smart city. However, secure context-awareness management in a trusted environment has not been considered in most of these research studies. As denoted above, a context-aware security system in a smart city could support the following requirements. First, it must ensure the security of context information exchange. To do so, a lightweight communication security scheme could be proposed. Second, the system must be able to assess the reliability of context information and reject unreliable context information. Finally, it must ensure trust management of context sources and context information coming from untrusted sources. A comparison of the studied work is presented in Table 1.

3. Context-Aware Security and Privacy as a Service

Our proposal integrates completely with the Context-Aware Security and Privacy as a Service (CASPaaS) architecture. Context-Aware Security and Privacy as a Service (CASPaaS) is a context-aware security and privacy architecture based on the "as-a-service" approach and enabling dynamic, flexible, and customized implementation of security and privacy services [7]. Thanks to the "as-a-service" approach, our architecture [7] allows the automatic composition of context-aware services. In addition, it enables the security and privacy support of generic IoT smart city applications through secure Application Programming Interfaces (APIs). CASPaaS has been designed to facilitate the integration of new network architectures and leverage their benefits in implementing context-aware security and privacy. Thanks to some new architectures/technologies (NFV: Network Function Virtualization, SDN: Software-Defined Networking, SFC: Service Function Chaining, etc.), CASPaaS can be considered as a service that can be placed at some strategic

Table 1: Comparison of significant context-aware security and privacy solutions in IoT.

Work	Context information security	Context information reliability	Context sources' trust mgmt.	Communication protocol	Artificial intelligence technique	Implemented and evaluated
[6]	No	No	No	Not specified	No	Yes
[2]	No	No	No	Not specified	No	No
[17]	No	No	No	Not specified	No	Yes
[19]	No	No	No	Not specified	No	No
[18]	Yes	No	No	Not specified	No	No
[22]	Yes	No	No	CoAP	No	Yes
[24]	Yes	No	No	MQTT	No	Yes
[25]	Yes	No	No	Not specified	No	No
[26]	No	No	Yes	Not specified	No	Yes
[27]	No	No	Yes	Not specified	Yes	Yes
[28]	No	No	Yes	Not specified	Yes	Yes
Our work	Yes	Yes	Yes	MQTT	Yes	Yes

nodes of the network infrastructure (e.g., edge nodes with an edge architecture). CASPaaS will thus be available at all times as close as possible to the users, while supporting their mobility.

CASPaaS is composed of two plans: the Knowledge Plan (KP) and the Security and Privacy Plan (SPP) (Figure 2). KP is responsible for the management of context awareness. Based on the context delivered by the KP, SPP implements the security and privacy mechanisms. SPP also has the role of ensuring the security of the architecture. To do so, it has several modules, including the Device Trust Management (DTM) module, which is responsible for the security management, the reliability of context information, and the trust of context sources. DTM transfers reliable context information to the KP's Context Acquisition (CA) module. CA performs the first processing on the received context information before storing it in the Context Information Base (CIB). The User Preferences Management (UPM) manages the user profile and preferences. More details could be found in [7].

4. Secure Context-Awareness Management in a Trusted Environment

In this section, we present the threat model, before detailing our proposal, and its two main mechanisms: the context information security and the trust management.

4.1. Threat Model. Implementing secure context-awareness management in a trusted environment requires prior analysis of various threats. These threats concern different levels:

(1) Context information exchange: in IoT context-aware security and privacy projects [2, 6, 14–16], context information is exchanged through clear communication channels. As a result, context information is vulnerable to eavesdropping attacks, which is easy to achieve under these conditions. Attackers can also intercept context information and exploit it: understanding the system, tracking the user without his knowledge, etc. It is also possible for attackers to replay captured context information or falsify it during its transmission.

(2) Devices: the system must not process context information provided by a compromised device. Threats to trust are numerous and include cloning, theft, spoofing, facilitated by the use of vulnerable firmware, and OTA (on the air) updates via unsecure channels.

(3) Trust management system: there are also several threats related to trust management systems, including attacks such as ballot stuffing, bad-mouthing, Sybil attacks, and selective behavior. Ballot stuffing and bad-mouthing attacks aim at increasing or decreasing the trust score of nodes. The identity change attack consists of changing the identity of a node that has received a low confidence score after malicious behavior so that its score can be reset. Thus, the trust management system has to be resistant to these attacks.

4.2. General Framework. The proposed solution protects the CASPaaS architecture against identity spoofing, eavesdropping, data tampering, and replay attacks. It also preserves privacy, thanks to the secure exchange of context information. As we will see in Section 5.2, the solution is also adapted to the smart city's constrained devices. Indeed, it has a low impact on energy consumption and offers better performance compared to existing solutions. Figure 3 illustrates the principle of secure exchange of context information where CASPaaS is placed in an edge infrastructure.

4.2.1. Implementation of Secure Exchange of Context Information. We propose to base communications on the MQTT protocol because it is well suited to meet the needs of collecting context information. Indeed, this protocol is lightweight and robust, supports QoS, and facilitates energy savings [34]. Thus, thanks to the pub/sub mechanism, IoT

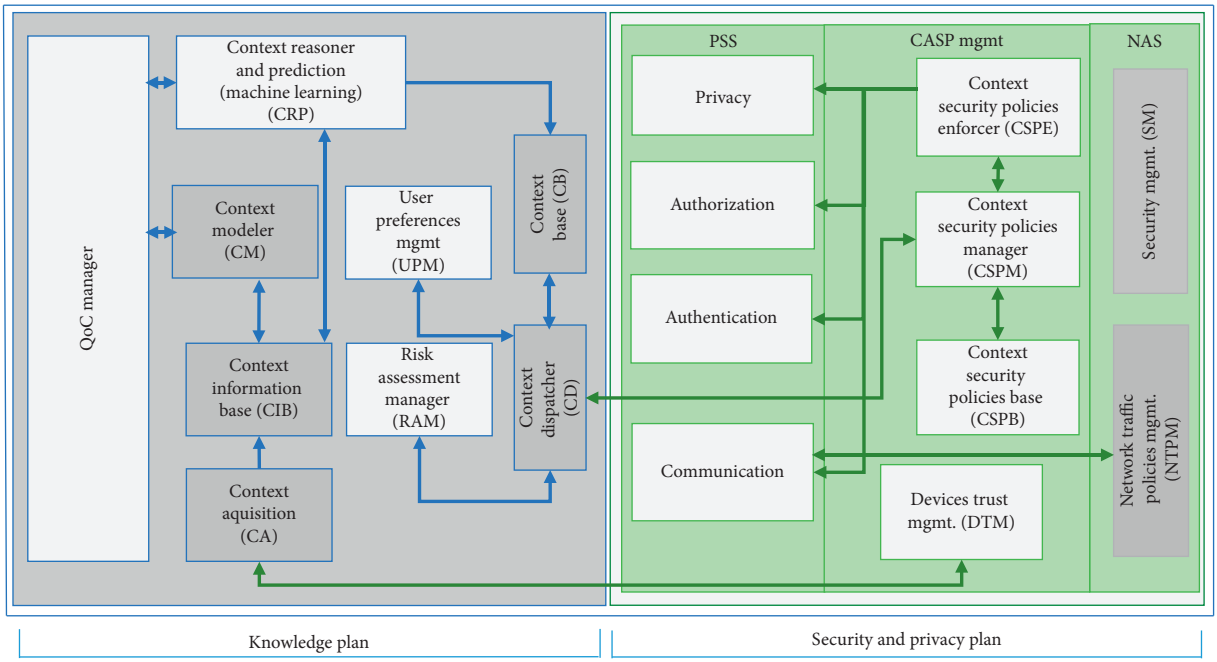


Figure 2: CASPaaS modules and their interactions [7].

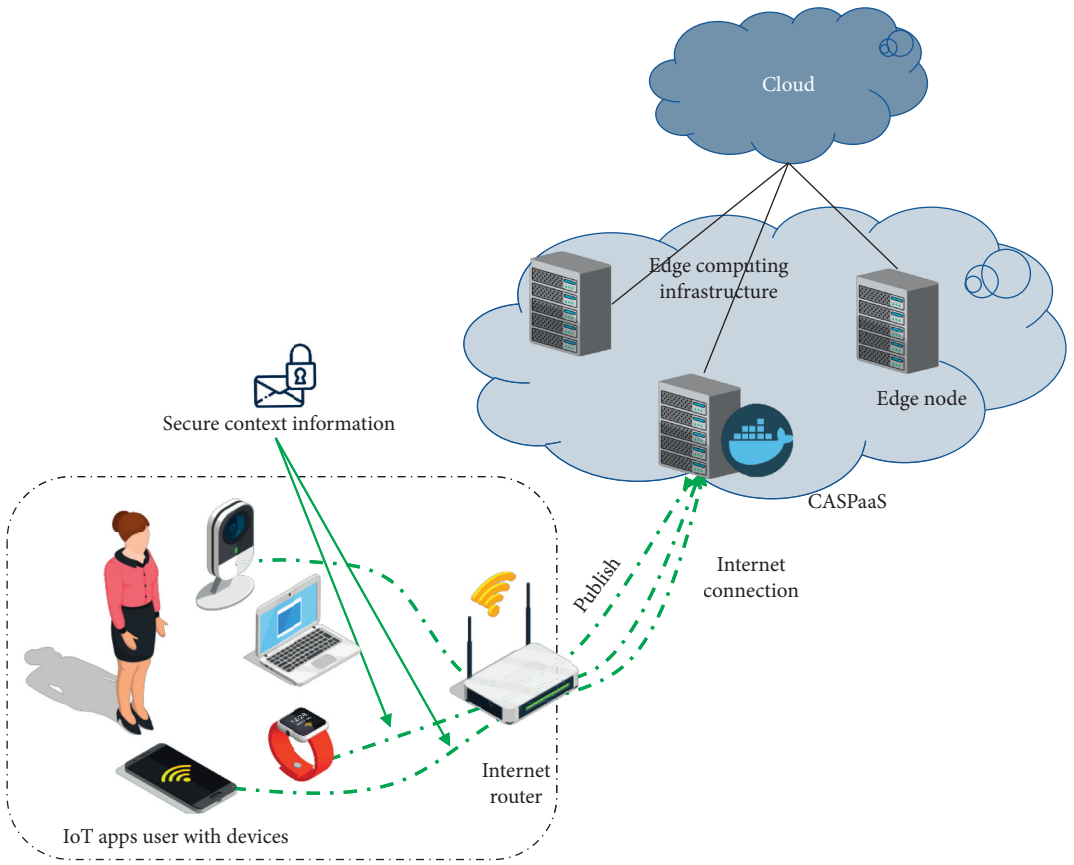


Figure 3: Secure collection and sending of context information to CASPaaS.

devices will send the context information to the broker. This context information acquisition process is shown in Figure 4. The broker handles the distribution of messages from the publishers to subscribers who have subscribed to a topic. For example, for a smart home application, the topic for information sent by a motion sensor might be home/room/motion.

The communication model is composed of three parts: the context sources (publishers), the CASPaaS Device Trust Management (DTM, subscriber), and the application broker. Context sources are constrained devices that collect, encrypt, and send secure context information to the DTM. The DTM receives, decrypts, evaluates reputation, and transfers reliable context information to the KP acquisition module (see Section 3). Specifically, when the DTM receives the information, its role is to verify this information before transferring it to the KP's context acquisition module. To do so, it evaluates the reputation of the source and transfers it to the KP acquisition module only if it is trusted. Assessments of the reputation of context sources and the reliability of context information are described in Section 4.3.

The MQTT protocol offers several methods to secure communications: SSL/TLS and payload encryption. The most widely implemented security mechanism for communication security in the MQTT protocol is SSL/TLS. This can be explained by the security and trust properties provided by SSL/TLS. In the IoT context, the management of certificates and session keys in SSL/TLS increases the computational complexity for the IoT devices. As a result, it leads to significant power consumption and a significant impact on their battery life. That is why we propose, in this work, to secure the application payload, which, as shown in Section 5.2, has a lower impact on the resources' consumption (power, CPU, memory, etc.). The payload includes the context information (e.g., latitude and longitude) and its subject (e.g., context).

The scientific contributions of this work are the following. First, we propose a new approach to secure communications based on MQTT. Indeed, this approach consists in securing the payload of the MQTT message. The major advantage of this approach is its ability to secure communications over unsecure networks. In addition, it is less resource-intensive than implementing TLS (the most used protocol for securing MQTT communications). Second, we show that this approach has a low impact on the transfer time of context information. It has a low impact on the energy and computing performance of constrained devices (cf. Section 5.2). Thus, it is possible to implement a secure exchange of context information without important impact on the resources' consumption for a context-aware security and privacy system.

The proposed system allows us to secure the exchange of context information, which is essential to protect our CASPaaS system against threats presented in Section 2. To do this, we need mechanisms that are lighter in terms of computing resources' consumption: CPU, memory, and thus energy. This need is justified by the constraints characterizing the context sources (IoT devices). Hence, the proposed system ensures confidentiality, data integrity, authentication, and no replay services. For confidentiality, we propose to use the AES (Advanced Encryption Standard) for the encryption of context information. Indeed, AES is a block cipher algorithm widely used to ensure communications' confidentiality in IoT. This is because most IoT devices have AES-specific cryptographic acceleration hardware. It also offers robust encryption with a reduced key size [35, 36] compared to an asymmetric encryption algorithm with the same robustness level. In addition, these encryption operations require fewer resources than asymmetric encryption. Furthermore, AES could be associated with other mechanisms to ensure integrity and authentication. Thus, we find many possibilities: Counter with CBC-MAC (CCM), Galois/Counter Mode (GCM), Electronic Codebook (ECB), etc. [37]. The most adapted cryptographic mechanisms for constrained devices are CCM and GCM [38]. In addition, AES-CCM is widely used because it has better security properties compared to AES-GCM [39, 40]. Thus, we propose AES-CCM to provide the communications of our CASPaaS with data integrity, authentication, and confidentiality.

However, since AES is a symmetric encryption algorithm using the same secret key for both data encryption and decryption, both parties must have the same key. Key exchange over unsecure networks faces several security issues. Indeed, during the key exchange phase, it is possible that an adversary intercepts the key. Furthermore, in most systems, keys are stored on devices, and these devices can be captured or cloned. Thus, to protect our system against these attacks, the keys used to encrypt context information should not be stored on the IoT devices. Therefore, implementing an algorithm allowing a secure exchange of single-use keys is necessary.

To solve the problems related to key exchange, we propose to use ECIES (Elliptic Curve Integrated Encryption Scheme) [41]. It is an authenticated public key cryptography system that aims at generating a secret key for onetime use by both parties of a communication. It combines Elliptic Curve Cryptography (ECC) and Diffie–Hellman primitive [42]. Using ECIES has several advantages in the secure management of context awareness compared to other lighter elliptic curve cryptography algorithms for IoT (e.g., Diffie–Hellman Elliptic Curve Cryptography). First, the use of ECIES does not require the use of a trusted third party. Then, the collection of context information must be performed in almost real time. So, ECIES has good performance and allows the use of onetime encryption keys. This enables avoiding the storage of keys on devices and therefore eliminates the threat of key reuse when a device is captured or cloned. The generated secret key is then used to encrypt data using AES-CCM (Confidentiality-Integrity-Availability: CIA; Keyed-Hash Message Authentication Code: HMAC). This data encryption ensures the confidentiality, integrity, and authentication.

In the following, we describe the proposed solution, which involves the following three steps: authentication, initialization of encryption mechanisms, and secure data exchange.

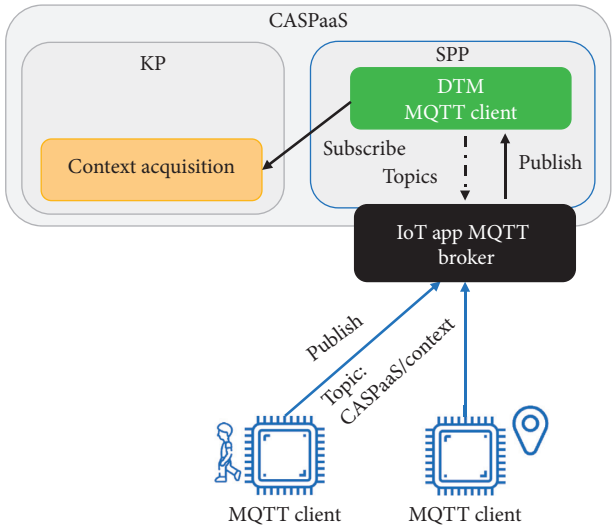


Figure 4: Context information acquisition with MQTT.

4.2.2. Authentication. This phase consists in authenticating different clients (i.e., the context sources) by the DTM. The MQTT protocol provides three authentication modes: client ID, username/password, and X.509 certificate. We propose the generation and use of a unique identifier per device. This identifier is derived from the cryptographic hash of the device's public key and its EUI-64 identifier. Indeed, the EUI-64 identifier based on the MAC address is unique and specific to each communication interface per device. The public key is also unique and can be generated on demand. Thanks to the robust cryptographic hash SHA-256, the generated unique identifier will be very difficult to reproduce [43]. A new public key will be periodically generated and the identifier recomputed. This will prevent attacks that could target the hash function and guess the public key/EUI-64 identifier association. The proposed unique identifier is defined as follows:

$$\mathrm{IDCS}_i = \mathrm{SHA} - 256\ (\mathrm{PkCS}_i + \mathrm{EUI} - 64\mathrm{CS}_i), \quad (1)$$

where SHA-256 is a cryptographic hash algorithm, IDCS_i is the context source unique identifier, PkCS_i is the public key of the context source, and $\mathrm{EUI\text{-}64CS}_i$ is the EUI-64 identifier of the context source.

The identifiers of the authorized context sources are previously registered with the DTM. If authentication is successful, the parties can proceed to the initialization phase. If authentication fails, the DTM notifies the trust management mechanism (see Section 4.2) and requests authentication again. After three unsuccessful attempts, authentication attempts are rejected for a period of 30 seconds (authentication rejection period). If another attempt fails, an additional 60 seconds is added to the authentication request rejection period, and so on. The rejection period spaces out authentication requests and reduces the risk of a denial-of-service attack.

4.2.3. Initialization. In the proposed system, the initialization consists in setting up ECIES by both the context source and the broker. This is equivalent to generating the security parameters of the ECIES algorithm. These parameters will generate the public/private key pairs that will be used for the key approval. In the following, the private and public keys of the context source will be, respectively, denoted SkCS_i and PkCS_i. Similarly, we will refer to the private and public keys of the DTM by, respectively, SkD and PkD.

When the authentication succeeds, the context source asks the DTM for the domain parameters to specify an elliptic curve and a reference point: $E(Fp)$, G, p, n, a, and b. These parameters must comply with the recommendations issued by NIST [44] and ANSSI [45]: ECC size of 256 bits, prime number of 2048 bits, etc. To do so, the broker generates a SkD private key and chooses a reference point G on the elliptic curve it has defined according to the security level. Then, he determines his public key PkD:

$$\mathrm{PkD} = \mathrm{SkD} \times G, \quad \mathrm{SkD} \in [1,\ p]. \quad (2)$$

The DTM sends PkD and the curve to the context source. Similarly, the context source CS_i generates SkCS_i and PkCS_i and sends PkCS_i to the DTM. SkCS_i represents the context source private key, and PkCS_i is the context source public key. PkCS_i is defined in the following equation:

$$\mathrm{PkCS}_i = \mathrm{SkCS}_i \times G, \quad \mathrm{SkCS}_i \in [1,\ p], \quad (3)$$

where $G = (x_G,\ y_G)$ is the base point on the finite field, generally noted Fp [46]. This secure public key exchange based on Diffie–Hellman key exchange is secure and takes place for the first time, i.e., when establishing the "secure channel" for context information exchange. It also takes place whenever it is necessary to reset the ephemeral keys of a device.

4.2.4. Secure Data Exchange. After the initialization phase, the secure sending of the collected context information takes place. To do so, the context source CS_i will need to create the shared secret key, noted Ssk, which will be used to encrypt the payload. First, this CS_i creates a shared random secret value R, resulting from scalar multiplication taking as inputs the private key of CS_i and the public key of the broker PkD.

$$R = \mathrm{SkCS}_i \times \mathrm{PkD}. \quad (4)$$

Then, the context source CS_i provides the shared random secret value R as an input parameter of the key derivation function (KDF). The KDF determines the shared secret key Ssk and the message authentication code computation key K_{MAC}. We propose to use a key derivation function based on HMAC-SHA-256. Indeed, the use of SHA-256 reduces the risk of success of brute-force attacks on the generated keys. This function outputs the concatenation of Ssk, the shared secret key, and K_{MAC}, the message authentication code computation key, each having a size of 128 bits. To do this, CS_i will use HMAC-SHA256, which is very secure and very difficult to "break." [47]

We propose to timestamp the payload to mitigate replay attacks. In this sense, we assume that each CS_i has a real-time clock that allows it to uniquely timestamp payloads. We also assume that, at each startup, the sources of context (CS_i)

synchronize their clocks with internet time. The antireplay mechanism consists in using the timestamps as unique, nonreproducible numbers generated during communications. The verification follows the following principle. CS_i authentication marks the beginning of context information exchange. The DTM uses the timestamp of the CS_i authentication as a time reference and starts a 300-second sliding time window for CS_i with a 30-second timeout. DTM uses this timeout (which can be set according to the requirements of the application to secure) to maintain the sliding time window or to stop it if CS_i does not perform any activity during this time. In the latter case, CS_i will need to authenticate itself to perform a new operation.

The 300-second sliding time window could be explained by the context update frequency, one of the major features of context-aware security and privacy. Indeed, the user's context can change frequently, requiring a high refresh rate. With such a high refresh rate, continuous authentication can reduce the energy autonomy of devices. This sliding window can reduce the power consumption associated with frequent authentication of the context source. Thus, at each CS_i publish operation, DTM validates the timestamp of the payload if it meets the following conditions:

(1) The timestamp of the payload is in the time window. If it is not the case, the payload is rejected.

(2) If this is the first publish operation after authentication, the timestamp must not be greater than the authentication timestamp by more than half a second. Otherwise, the payload will be rejected because it will be considered too old. The delay of half a second, i.e., 500 milliseconds, is justified by the tolerance to latencies that can be caused by the disruptions of the used access networks.

When the DTM validates a payload, it updates the CS_i time window using the last validated timestamp as the lower bound and extends the upper bound to 300 seconds. When CS_i does not perform any operation, after the timer expiration, it must authenticate, and a new time window must be set up. Thus, the DTM maintains a sliding time window of timestamps already validated by CS_i and rejects all payloads that have a timestamp already validated and or outside the window.

The context source CS_i uses the shared secret key Ssk to encrypt the timestamped payload using 128-bit AES-CCM. The result of this operation is an encrypted message, denoted M_{ENC}. From this M_{ENC} and the K_{MAC} key, CS_i uses the HMAC-SHA256 function to compute a MAC tag. Finally, this CS_i sends the couple (M_{ENC}, tag) to the broker. Figure 5 illustrates the structure of the MQTT data packet formed by CS_i.

When the DTM receives the encrypted payload, it extracts the pair and does the reverse process using the CS_i's public key. To do so, it recomputes the shared secret key using the key derivation function and the parameters previously established with the context source. Thus, it computes a tag' and compares it to the tag sent by CS_i. If tag' and tag are different, it aborts the process. If tag' and tag are equal, it proceeds to M_{ENC} decryption. It checks the timestamp of the payload to verify its validity. If the timestamp verification fails, it rejects the packet. The PUBLISH operation is defined by the following equation:

$$\mathrm{PUBLISH}\left(\mathrm{ECIES\_Key\_Exchange}\left(G_i\right), \mathrm{Encrypt\_AES-CCM}\left(\mathrm{CI}, \mathrm{Ssk}, \mathrm{Tp}, \mathrm{ts}, \mathrm{IDCS}_i\right)\right), \tag{5}$$

where CI is the context information, Ssk is the shared secret key, Tp is the topic, and ts is the timestamp.

Once the message is validated, DTM checks the trust index (TI) of the context source using the trust management mechanism. If the source is trustworthy, it then verifies the reliability of the context information using the context information reliability management mechanism. Depending on the result, the received context information is transferred to the context acquisition module or rejected (see Section 4.3). A summary of the operation of secure transmission of context information by a context source is presented in Figure 6.

4.3. Context Sources' Trust Management. In this section, we present the context sources' trust management mechanism. This mechanism uses context information reliability and context source behaviors to assess the reputation of context sources. By assessing the reliability of context information and the behavior of context sources, the system can detect false context information.

4.3.1. Overview. Context source trust management allows to manage trust relationships with context sources. It defines how to establish, maintain, or revoke a trust relationship with a device. The goal is to allow the context-awareness management system to handle only reliable context information provided by trusted context sources. This protects the CASPaaS architecture against erroneous or inappropriate adaptation decisions.

There are several models for managing trust relationships in IoT. Among these models, we can cite negotiation, reputation evaluation, and predefined policy decisions [48]. The choice of a model is based, on the one hand, on the interaction model of different nodes and, on the other hand, on the data coming from these interactions. According to these elements, the reputation model approach is well suited for a context-aware security and privacy environment in the IoT. Indeed, in this environment, we can have the data (reliability of context information and device behavior in our case) to evaluate experiences of context sources. Reputation can be considered trusting or not based on experiences and/or observations, whether good or bad. Thus, our trust

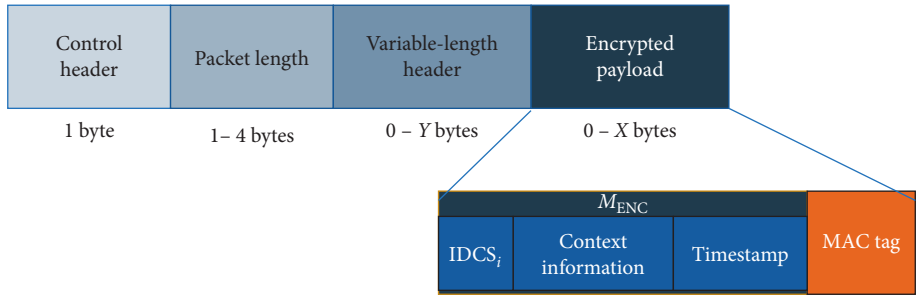


Figure 5: MQTT packet with a secure context information payload.

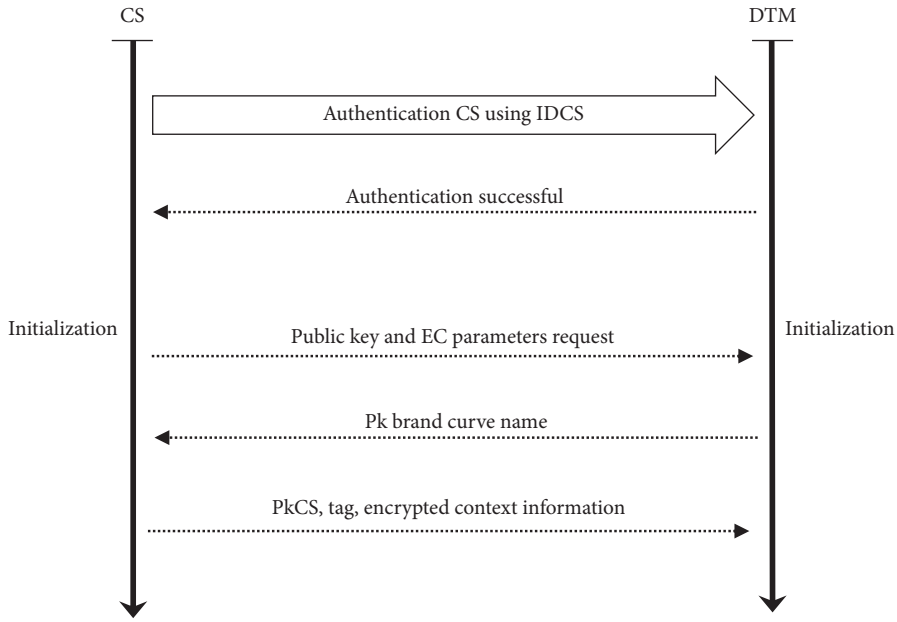


Figure 6: Sequence of secure context information exchange.

management system will be based on the reputation evaluation model.

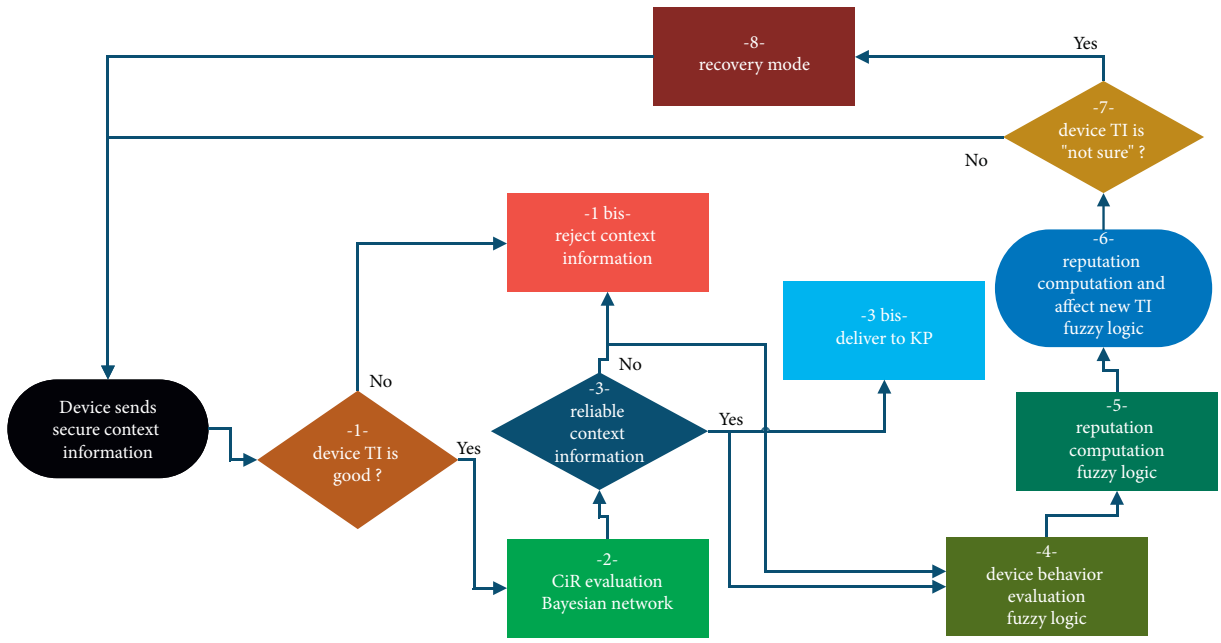
This reputation evaluation is based on the dynamic assessment of context information reliability and context source behavior. To do so, the context information reliability is evaluated using a Bayesian network. The context sources' behavior is evaluated according to the feedback from the MQTT broker (connection attempts, multiple authentication attempts, etc.) and the DTM (e.g., unauthenticated message). Figure 7 illustrates the proposed reputation assessment process.

4.3.2. Reputation Evaluation. The mechanism for evaluating the reputation of context sources is based on the devices' behaviors and the reliability of context information they provide (Figure 7).

Specifically, there are three steps in the reputation evaluation of a context source. The following sections describe these steps.

(1) Evaluating the Reliability of Context Information. As introduced in Section 4.3.1, the proposed reputation management system is based in part on the evaluation of context information reliability for assessing the reputation of a context source. Evaluating the reliability of context information in a context-aware security and privacy environment in the smart city involves establishing the consistency of this context information with the user's actual context. It allows confirming or invalidating context information depending on the quality of sensors, the presence of other sensors, and information about the user (e.g., habits and agenda content).

There are several verification methods, including the comparison of context information from the considered source with other context information from other context sources and with user profile information. Artificial intelligence is well suited for assessing the credibility of context information and detecting suspicious activities [49, 50]. Several artificial intelligence techniques could be used for verification and validation: linear regression, support vector machine, decision tree, neural networks, etc. [51]. The choice of a technique depends on the type of learning (supervised and unsupervised), the amount of input data needed, and the accuracy of these data. The information available in our environment (smart city) is not important, and it is often



CiR : Context Information Reliability
TI : Trust Index

Figure 7: Proposed reputation evaluation process.

uncertain. For example, by providing context information (geographical position, date, and time), we should be able to know if this geographical position is consistent. In this case, there are few input data available, and these data can be uncertain in some cases. Thus, a Bayesian network is well adapted to our environment. Indeed, the advantage of Bayesian networks is that they make it possible to solve problems with a limited amount of uncertain data [52].

Using a Bayesian network, a Bayesian network is a directed acyclic graph (DAG) in which the nodes represent the random variables and the arcs represent the correlation probabilities between these variables [53]. It models uncertainties and calculates these uncertainties using the concept of probabilities. The network obtained after modeling a decision-making problem by a Bayesian network represents a joint probability distribution.

In addition, the vast majority of IoT applications in the smart city are based on the user's profile in order to offer intelligent and personalized services. Thanks to the user profile information available in these applications, it will be easy to correlate context information with this profile to determine its reliability. Indeed, as denoted by Schiaffino, context is an element of the user profile [53]. In fact, the context information of the user's profile is built up from past observations of the user's contexts [50]. A context is characterized by context information allowing to determine a location (home, work, shopping mall, sports halls, etc.) and an activity performed by the user (rest, walking, sleep, sports, driving, etc.) at a specific time (time, day, date, etc.). Thus, the presence of the user in a place can be determined by his profile. In other words, this presence can be determined by the joint conditional probabilities of the geolocation, the time (hour and day of the week), and the used network. The presence can be reinforced by the user's agenda.

Our approach therefore consists in using a Bayesian network (Step 2) to combine the context information received with the user's other profile information (e.g., the network, the activity, and the agenda) in order to determine the probability of reliability of this context information. The usefulness of the interdependence of the user's profile and his context is that it allows the validation of the context information transmitted by the current context sources from those stored in the user's profile. Therefore, with a well-informed user profile (e.g., habits, frequented places, time and day of frequentation, activities carried out, and access networks), the interdependence between the user profile and the contexts makes it possible to reinforce the credibility of the context information transmitted by the context sources. Figure 8 represents the knowledge network of the evaluation process we have defined. From this information, inferred random variables and determined conditional probabilities allow us to build a Bayesian network which is able to determine the probability that context information transmitted at a given time (e.g., GPS position) is reliable or not. Initially, the context information is considered reliable (Step 3) if the probability determined by the Bayesian network is greater than or equal to a certain threshold, the value of which will be set at 80% following the results of experiments detailed in Section 5.3. Otherwise, it is not reliable. This threshold is defined in order to minimize the number of false positives resulting from this evaluation. The Bayesian network will learn from the user profile evolution and contexts

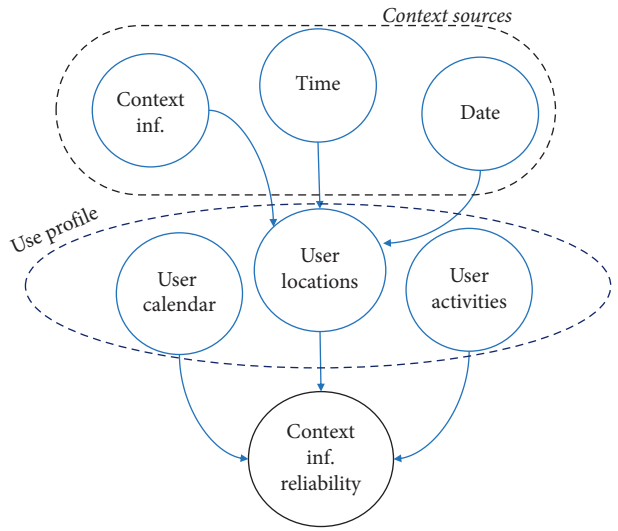


Figure 8: Bayesian network for the evaluation of context information reliability.

determined by the KP. The User Preferences Module (UPM) of the CASPaaS architecture handles the user profile.

The joint probability distribution of all variables is presented in the following equation:

$$P(\text{rci},\ \text{cal},\ \text{loc}, \text{act}, \text{cinf}, d, t) = P(\text{rci}|\text{cal}, \text{loc}, \text{act}) \times P(\text{cal}) \times P(\text{loc}|\text{cinf}, d, t) \times P(\text{act}) \times P(\text{cinf}) \times P(d) \times P(t), \quad (6)$$

with cal: agenda, loc: location, act: user's activity, d and t: date and time of context information observation cinf, and rci: probability of context information reliability.

Evaluating the conditional probabilities of each random variable yields

$$P(\text{rci},\ \text{cal},\ \text{loc}, \text{act}, \text{cinf}, d, t) = \frac{P(\text{rci}, \text{cal}, \text{loc}, \text{act})}{P(\text{cal}, \text{loc}, \text{act})} \times P(\text{cal}) \times \frac{P(\text{loc}, \text{cinf}, d, t)}{P(\text{cinf}, d, t)} \times P(\text{act}) \times P(\text{cinf}) \times P(d) \times P(t). \quad (7)$$

If the probability of context information reliability rci is under the reliability threshold, then the context information is rejected (Step 1-bis). This may result from false context information, possibly forged.

For example, it is likely that Bob is in the office on Thursday between 8 am and 5 pm, so the joint probability of the user's presence at the office, knowing the time (day of the week and hour), the GPS position, and the WiFi network on which his phone is connected, can be close to 100%. The accelerometer and the pedometer rarely change much during this time frame because Bob, being seated at his desk most of the time, makes few movements. In addition, Bob has scheduled a work meeting on his agenda. These data support the probability that Bob is actually at the office (day, hour). However, Bob had his connected watch stolen two days ago. The context-aware security and privacy system receives the GPS location from Bob's connected watch. This position indicates a location away from Bob's home and office. When this GPS position, day of the week, and time are passed to the Bayesian network, the probability of reliability of this information can be between 3 and 10%. Thus, with a well-constructed and well-trained Bayesian network, it becomes easy to detect false context information. Therefore, our solution can detect context information forged or coming from cloned or stolen devices.

(2) Context Sources' Behavior. The MQTT protocol has several connection control packets: CONNECT, CONNACK, PUBACK, etc. Through these packets, we propose to evaluate (Step 4) the exchanges in order to detect suspicious behaviors of different context sources (multiple connection attempts, multiple authentication attempts, abnormally large message, unknown topic, etc.). The proposed mechanism uses fuzzy logic to detect the behaviors of the context sources at the broker level, according to the device activities.

Using fuzzy logic: fuzzy logic is an artificial intelligence technique allowing to produce interesting reasoning from uncertain data [54]. It is flexible, requires few data, and tolerates their imprecision. Indeed, computations are based on IF-THEN rules.

In a specific way, the proposed mechanism produces statistics that provide indicators on the behavior of a source. These indicators include, for instance, the rates of connection requests, authentication failures, and messages with abnormal size. Thus, we argue these indicators can help to detect the behavior of a context source. Indeed, a high rate of aborted connection attempt can indicate that a source is compromised and is conducting a denial-of-service attack against the broker. A high authentication failure rate may indicate that a malicious source is trying to authenticate without success. An unusually large message size may indicate that a source is trying to degrade the broker's performance. Similar work has been proposed in [55]. However, the indicators and processes used in that work differ from ours. Indeed, the authors used connection request and connection acknowledgment rates for a denial-of-service attack detection system. On the contrary, our indicators allow us to quickly determine the behavior of a context source.

Therefore, these proposed indicators will be fuzzified and provided as the input to our fuzzy inference system. Each fuzzy variable (indicator) includes the fuzzy sets: low and high. The low set contains the values indicating a low ratio of the considered indicator (connection request rate, authentication failure rate, and abnormal message rate). The high set contains values indicating a high ratio of a given indicator. For example, the authentication failure rate can be considered low when it is below 10% and high when it is above 10%. The connection request rate is different and can be considered high when it is above 40% and low when it is below 40% [34].

In fuzzy logic, the limit values for belonging to a fuzzy set are not precisely defined [54]. They depend on the fixed objectives. Thus, we can have different values for low or high levels. We estimate the possibilities in trust for different indicators. A node that sporadically or randomly scans the broker will have a higher connection request failure rate compared to a node that performs legitimate operations. Similarly, a node that aims at taking the broker out of service will have a higher connection request failure rate. Thus, the connection request failure rate is 100% low when the ratio is between 0 and 30%. From 25 to 100%, the probability of the high rate increases, while the probability of low rate decreases.

We argue that the authentication failure rate of a context source is low when it is less than 20%. Indeed, one or two failures for every ten authentications is not indicative of a malicious behavior. On the contrary, a failure rate higher than 30% may indicate that the context source is performing a brute-force attack or a man-in-the-middle attack (MITM). The rate of abnormally large messages is the same as the authentication failure rate. In the following, we provide the formulas enabling the evaluation of different rates and their membership functions.

ConnRateS_i indicates the rate of connection requests for a CS_i context source:

$$\text{ConnRateS}_i = \frac{\text{ConnectPacketS}_i}{\text{TotalConnectPackets}}, \tag{8}$$

with ConnectPacketS_i, the number of CONNECT packets sent by the CS_i source, and TotalConnectPackets, the total number of CONNECT MQTT packets observed over a period. The membership features of ConnRateS_i are

$$\mu_{\text{low}}\left(\text{ConnRateS}_{\text{i}}\right) = \begin{cases} 0, & \text{ConnRateS}_i \le 0, \\ \dfrac{\text{ConnRateS}_i}{15\%}, & 0 < \text{ConnRateS}_i \le 15\%, \\ \dfrac{30\% - \text{ConnRateS}_i}{30\% - 15\%}, & 15\% < \text{ConnRateS}_i < 30\%, \\ 0, & \text{ConnRateS}_i \ge 30\%, \end{cases} \tag{9}$$

$$\mu_{\text{high}}\left(\text{ConnRateS}_i\right) = \begin{cases} 0, & \text{ConnRateS}_i \le 25\%, \\ \dfrac{\text{ConnRateS}_i - 25\%}{50\% - 25\%}, & 25\% < \text{ConnRateS}_i \le 50\%, \\ \dfrac{100\% - \text{ConnRateS}_i}{100\% - 50\%}, & 50\% < \text{ConnRateS}_i < 100\%, \\ 0, & \text{ConnRateS}_i \ge 100\%. \end{cases} \tag{10}$$

$\mu_{\text{low}}(\text{ConnRateS}_i)$ and $\mu_{\text{high}}(\text{ConnRateS}_i)$ are, respectively, the membership function of low and high connection rate of a context source.

The variable FARateS_i (equation (11)) represents the rate of failed authentications of a CS_i context source:

$$\text{FARateS}_i = \frac{\text{NFailedAuthS}_i}{\text{NAuth}}. \tag{11}$$

NFailedAuthS_i is the number of authentications of the context source, and NAuth is the total number of authentications, i.e, from all context sources observed over a period.

$$\mu_{\text{low}}(\text{FARateS}_i) = \begin{cases} 0, & \text{FARateS}_i \le 0, \\ \dfrac{\text{FARateS}_i}{10\%}, & 0 < \text{FARateS}_i \le 10\%, \\ \dfrac{20\% - \text{FARateS}_i}{20\% - 10\%}, & 10\% < \text{FARateS}_i < 20\%, \\ 0, & \text{FARateS}_i \ge 20\%, \end{cases} \tag{12}$$

$$\mu_{\text{low}}(\text{FARateS}_i) = \begin{cases} 0, & \text{FARateS}_i \le 18\%, \\ \dfrac{\text{FARateS}_i - 18\%}{40\% - 18\%}, & 18\% < \text{FARateS}_i \le 40\%, \\ \dfrac{100\% - \text{FARateS}_i}{100\% - 40\%}, & 40\% < \text{FARateS}_i < 100\%, \\ 0, & \text{FARateS}_i \ge 100\%. \end{cases} \tag{13}$$

$\mu_{\text{low}}(\text{FARateS}_i)$ and $\mu_{\text{low}}(\text{FARateS}_i)$ are, respectively, the membership function of low and high failed authenticate rate of a context source.

Finally, AMSRS_i (equation (14)) indicates the rate of abnormally large messages sent by a CS_i context source:

$$\text{AMSRS}_i = \frac{\text{AMSS}_i}{\text{NMS}}, \tag{14}$$

with AMSRS_i, the number of abnormal-size messages sent by CS_i, and NMS context source, the total number of normal-size messages observed over a period.

$$\mu_{\text{low}}(\text{AMSRS}_i) = \begin{cases} 0, & \text{AMSRS}_i \le 0, \\ \dfrac{\text{AMSRS}_i}{10\%}, & 0 < \text{AMSRS}_i \le 10\%, \\ \dfrac{20\% - \text{AMSRS}_i}{20\% - 10\%}, & 10\% < \text{AMSRS}_i < 20\%, \\ 0, & \text{AMSRS}_i \ge 20\%, \end{cases} \tag{15}$$

$$\mu_{\text{low}}(\text{AMSRS}_\text{i}) = \begin{cases} 0, & \text{AMSRS}_i \le 18\%, \\ \dfrac{\text{AMSRS}_i - 18\%}{40\% - 18\%}, & 18\% < \text{AMSRS}_i \le 40\%, \\ \dfrac{100\% - \text{AMSRS}_i}{100\% - 40\%}, & 40\% < \text{AMSRS}_i < 100\%, \\ 0, & \text{AMSRS}_i \ge 100\%. \end{cases} \tag{16}$$

$\mu_{\text{low}}(\text{AMSRS}_i)$ and $\mu_{\text{low}}(\text{AMSRS}_i)$ are, respectively, the membership function of low and high abnormal message size rate sent by a context source.

The thresholds of the membership functions defined in equations (9), (10), (12), (13), (15), and (16) are explained in Section 4.3.2. (2).

The membership functions of indicators are, respectively, presented in Figures 9(a)–9(c).

For each rule in the rule base, an appropriate implication has to be applied. Each implication is composed of an antecedent and a consequence. The result of the implication rule is then aggregated and defuzzified to obtain the result. This result will be used by the context source reputation evaluation mechanism to determine the trust index of the context source. This mechanism is described in Section 4.3.2.3. The input fuzzy variables are ConnRateS_i, FARateS_i, and AMSRS_i. The fuzzy inference system based on the Mamdani model uses fuzzy rules to determine the behavior of each context source. Table 2 shows the rules for device behavior. The output of the inference system is also a fuzzy variable. It is defuzzified in order to get the nonfuzzy values (crisp values) representing the resulting decision of the process. As a final evaluation of the device behavior, we have the following values: good, doubtful, and malicious.

(3) Context Sources' Reputation Management. When the system is initialized, the context sources have the maximum trust index (TI) (Step 1). This index is evaluated each time the context information is provided. When the DTM receives and decrypts the context information, it starts the context source reputation evaluation cycle (Figure 7). To do so, the reliability of the information received and the behavior of the context source are successively evaluated. If the context information is not reliable, then it is rejected (Step 1-bis), and the reputation evaluation process continues to determine the trust index (TI). In Step 2, the system evaluates the reliability of the context information (see Section 4.3.2.1). In the next step, it determines the device behavior (see Section 4.3.2.2). Once the behavior of the device has been determined and the reliability of the context information has been evaluated, its trust index is computed by the reputation management mechanism (Steps 5 and 6).

The trust index provides a direct indication of whether or not a context source is trustworthy, on a scale of 0 to 1. Table 3 summarizes the TI values' range. At each reputation evaluation (Step 7), the value of the trust index is increased, decreased, or remains unchanged. When the value of the trust index reaches the "not sure" level, the system sends a notification to the user and puts temporarily the device in the recovery mode (Step 8). In this temporary mode, the context information is evaluated but is not delivered to the KP. This mode resets the trust level of a device with the support of user's feedback. The recovery mode can be used when a device has been recovered after theft or when a device has sensors that need to be recalibrated. However, when a device is put into the recovery mode more than two times, it may be a sign that it is compromised. In this case, the system notifies the user that the device is no longer safe and should be removed from context sources. At this point, the user can, if possible, perform a hardware reset of that device and add it back to the system.

The proposed mechanism computes the trust index using fuzzy logic. The evaluation of the trust index involves uncertainty because it is based in part on an uncertain and fuzzy element that is the behavior of the device. Thus, fuzzy logic is well adapted to this case because it allows to deal with uncertainties that cannot be strictly treated with the likelihood of probability. As a result, the input fuzzy variables are context information reliability and context source behavior. The reliability of context information has the following fuzzy sets: not reliable, doubtful, and reliable. These elements are the outcomes of the context information reliability evaluation mechanism. The behaviour of the context source has the following fuzzy sets: the output of the context source behavior evaluation mechanism (good, doubtful, and malicious). The membership functions characterizing the reliability of context information and the behavior of the context source are shown in Figures 10(a) and 10(b).

Although the reliability of context information and the behavior of context source devices are two different information, we combine them to get the trust index. This is done through the use of a rule base following a Mamdani fuzzy inference system. Table 4 represents the rules used by the fuzzy inference system that we defined. The result represents the trust index and therefore the trust level of the context source: sure, doubtful, faulty, and not sure (compromised). As with the behavior evaluation mechanism, the rule base is expressed as "if-then, if not." For example, if "the context information is reliable" and "the behavior is good," then "the trust level is sure." On the contrary, if "the context information is doubtful" and "the behavior is good," then "the trust level is faulty." In all cases, the reputation of the context source is evaluated to within one previous action. This will allow the context-aware security and privacy system to discard compromised context sources.

Since the defined system is centralized and sources do not evaluate each other, this system is resistant to ballot stuffing and bad-mouthing attacks. In addition, the identities of the context sources cannot be spoofed (see Section 4.2.1). Thus, the proposed system is resistant to identity change attacks.

5. Performance Evaluation

In this section, we analyze the performance of our proposal: SETUCOM. First, we describe the implementation and experimentation conditions. Second, we compare the key performances of SETUCOM and SSL/TLS. After that, we analyze the performance of the proposed reputation management system. Finally, we evaluate the malicious detection rate.

5.1. Experiment Setup. In our simulation, we considered a user of a smart home application with three IoT devices as context sources: a connected smart watch, a connected pedometer (step counter), and a smartphone. These devices are based on Raspberry Pi Zero W having a Broadcom

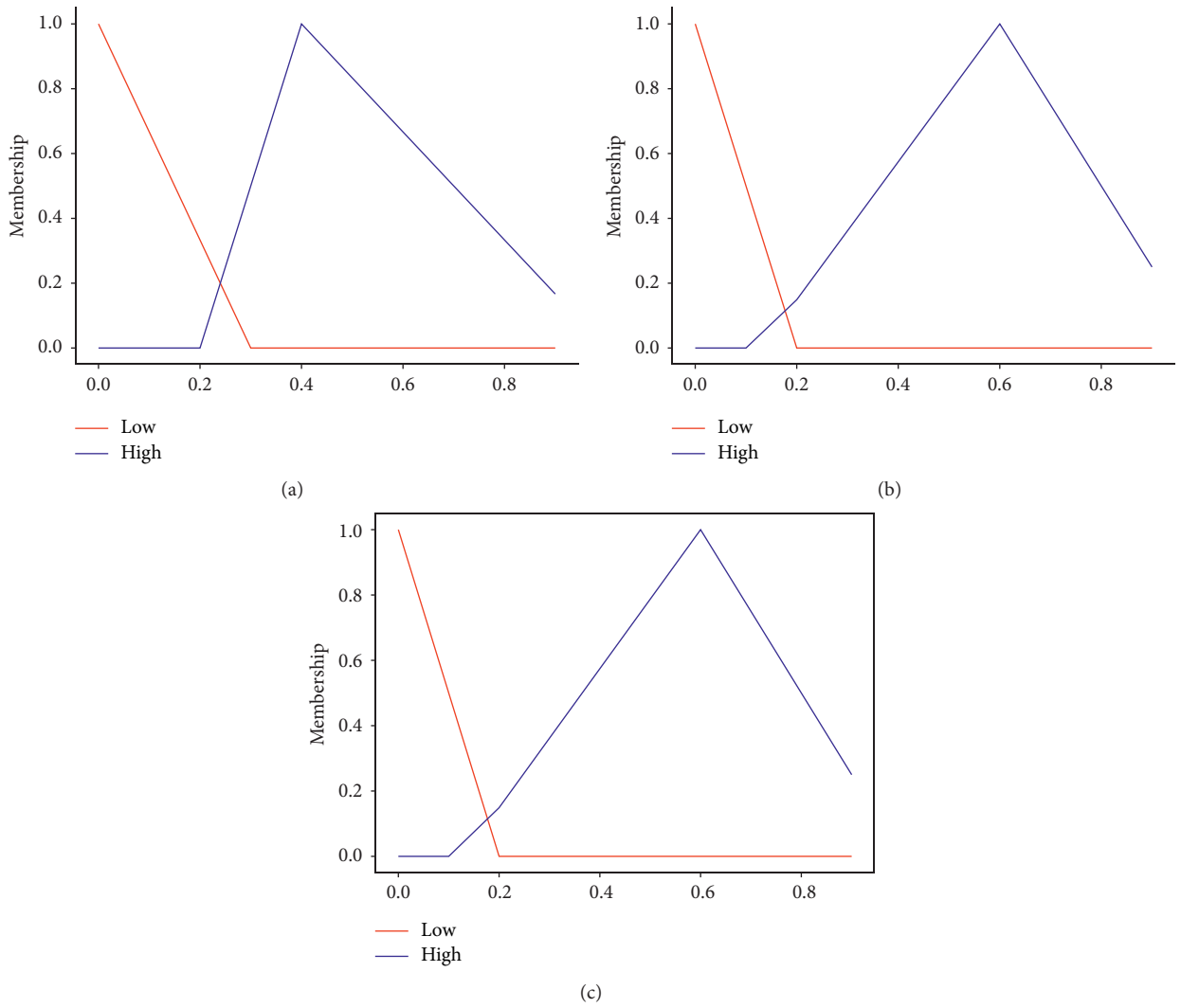


Figure 9: Behavior indicators' membership functions: (a) connection attempt rate, (b) failed authentication rate, and (c) abnormal message rate.

Table 2: Context source behavior rules' table.

Index	Failed connection ratio	Failed authentication ratio	Abnormal message ratio	Device behavior
1	Low	Low	Low	Good
2	Low	High	Low	Malicious
3	Low	Low	High	Doubtful
4	Low	High	High	Malicious
5	High	Low	Low	Doubtful
6	High	High	Low	Malicious
7	High	Low	High	Malicious
8	High	High	High	Malicious

Table 3: Trust index values' range.

Trust level	Trust index
Sure	$TI \geq 0.8$
Faulty	$0.6 \leq TI < 0.8$
Doubtful	$0.4 \leq TI \leq 0.6$
Not sure	$TI < 0.4$

BCM2835 chip based on the 700 MHz ARM1176 processor and a 256 MB memory. They are configured with the Raspbian Buster system and have Wi-Fi connectivity. The DTM is hosted as a service on a Dell computer configured with Ubuntu 18.04 LTS (64-bit), Intel Core i7 vPro 5th generation Dual Core 2.60 GHz, and 12 GB memory.

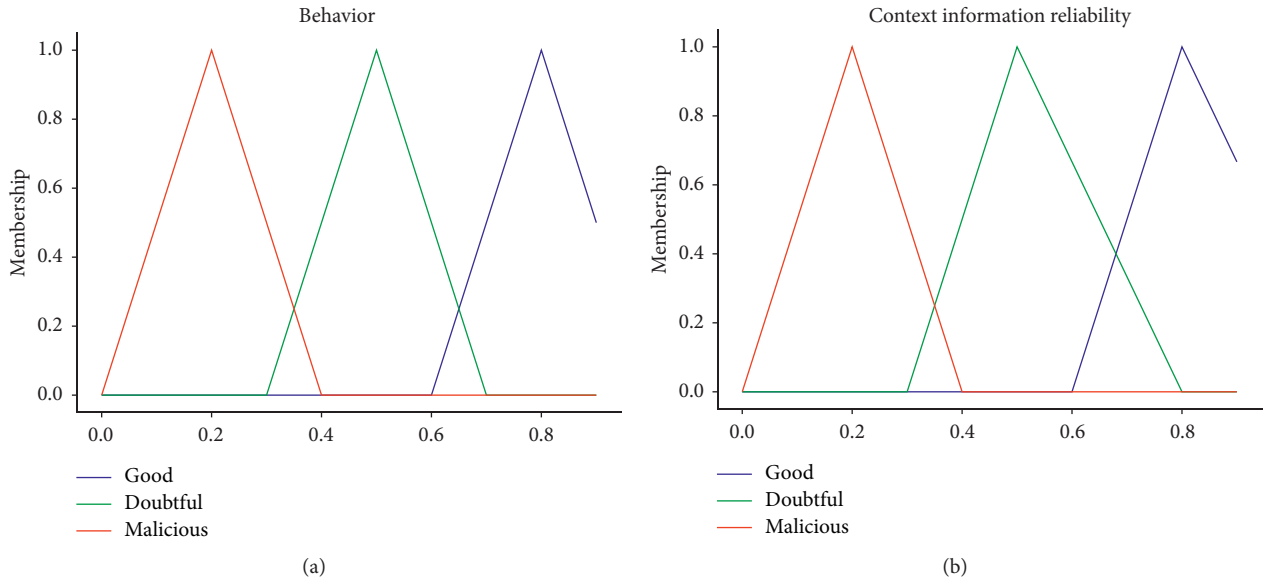


Figure 10: Reliability and behavioral membership functions.

Table 4: Inference rules for trust level evaluation.

Index	Context information reliability	Context sources' behaviors	Trust level
1	Reliable	Good	Sure
2	Reliable	Doubtful	Faulty
3	Reliable	Malicious	Doubtful
4	Doubtful	Good	Faulty
5	Doubtful	Doubtful	Doubtful
6	Doubtful	Malicious	Not sure
7	Not reliable	Good	Faulty
8	Not reliable	Doubtful	Not sure
9	Not reliable	Malicious	Not sure

The MQTT client used at the context-source level is based on the Paho MQTT Embedded C-library, a lightweight open-source library widely used in IoT research and industry [56]. We have written the MQTT client in C++. Thus, we have used the CryptoPP library [57] to write the cryptographic algorithm to ensure the secure exchange of context information. The MQTT broker used is version 3.1.1 of Mosquitto, an open-source broker, also widely used [58] and written in C++/Python. We used the pyAgrum library to develop the Bayesian network for the detection of the reliability of context information [59]. It is a powerful library written in C++ and adapted in Python. We also use the Python language to write fuzzy logic algorithms through the skfuzzy library [60]. skfuzzy is a powerful open-source library written in Python and allowing to create complex fuzzy logic algorithms.

5.2. Comparison of the Overload of SSL/TLS and Our Solution. SSL/TLS is the most used protocol for securing IoT communications based on the MQTT protocol. However, the overload induced by this protocol is not acceptable for most IoT devices, not only due to the processing time but also due to the energy consumption. We demonstrate the feasibility of our solution and point out its advantages in terms of execution time and memory usage in constrained devices. Thus, we compare our proposal with a context-aware system that does not implement secure exchanges of context information and a system using TLS for securing these exchanges. For this, we use the Paho MQTT C client for the IoT with TLS mutual authentication and a 2048-bit certificate.

The overload required for each payload, therefore for each packet sent with the proposed system, is 40 bytes compared to the same packet sent without security and having an average size of about 100 bytes. This is explained by the addition of the authentication data through AES-CCM encryption with a fixed size of 8 bytes and the tag (HMAC) authenticating the message also having a fixed size of 32 bytes. The large part of the overload size is proportional to the size of the public key. The overload will have approximately the same size, regardless of the size of the context information to be sent. It is acceptable and requires only 40 bytes of additional data.

Figure 11 illustrates the processing time for messages sent with the proposed system, compared to a system without security and a secure system with SSL/TLS. The overload of the proposed system is acceptable compared to

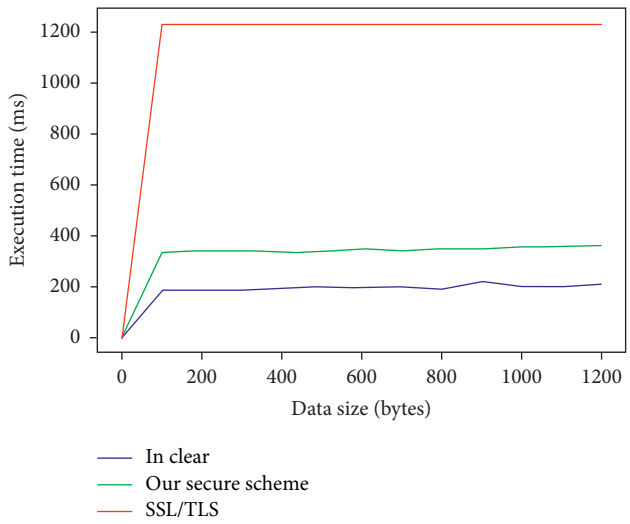


Figure 11: Average time to send context information.

that induced by SSL/TLS security. The use of the SSL/TLS protocol requires session establishment (i.e., the implementation of the TLS Handshake at each publish operation). This includes the ClientHello and ServerHello messages of an average of 250 bytes, the exchange and verification of certificates, an average of 3000 bytes, and key exchange. Thus, these initial exchanges generate an important overload that our solution allows to reduce (we have the unique sending of the public key during the first exchange).

Furthermore, the Handshake of the TLS protocol is resource intensive and therefore energy consuming, unlike the system we propose. Thus, sending context information with SSL/TLS requires an average of 1230 ms. Compared to SSL/TLS, the system that we propose allows a gain of about 900 ms. Compared to a system without security, the proposed system has an additional execution time of about 160 ms. Compared to SSL/TLS, this delay is acceptable because it has a small impact on the overall time of context-awareness management, including the time of collection and transmission and the processing time of the context information. A context-aware security system must process context information very quickly.

Concerning the used memory, our system uses at most about 4000 kbytes. It is the same for a context information collection program that does not implement the security of these exchanges. In contrast, the Paho MQTT C client using SSL/TLS consumes nearly 10% more memory than our solution. Figure 12 illustrates the obtained results. The additional memory consumption of the Paho MQTT C client is due to the excessive consumption of resources required to initialize TLS-based exchanges. Indeed, TLS significantly affects performances, especially processor usage during the exchange phase. In addition, it has a considerable impact on the energy consumption of devices. Thus, the system proposed in this paper is really efficient (processor, memory, and energy). Figure 13 illustrates the respective average energy consumption of the three systems during their respective average execution times of 186 ms, 300 ms, and 1210 ms.

5.3. Context Sources' Trust Management. The first step of the proposed trust management system is the evaluation of context information reliability. This mechanism, based on a Bayesian network (Figure 8), needs only some information for a good reliability evaluation. In our simulation, we considered the geographical position and the motion speed as primary context information provided by the user's devices. In some cases, information from the profile such as the user's calendar and/or routine activities can be used to increase the accuracy of the evaluation. To conduct this experiment, the network was trained with more than 1400 joint probability conditions built from Bob's simplified habits.

For example, we reconsider the Bob case (Section 4.3.2.1). It is Thursday at 10:00 am., and according to his profile, Bob is supposed to be at work (office). When Bob is in his office, he makes almost no movement because he remains seated most of the time. Bob's smart watch, which is the source of context, sends the geographical position and the speed of his movement. After extraction of the geographical area by the geofencing technique, the determined area is the workplace. The speed of movement denoted m and the geographical zone denoted p are transmitted to the system. The data of the calendar denoted c and the usual activity denoted a are not known in this example. Therefore, they have default values, respectively, nothing and unknown. Figure 14 illustrates the result of the evaluation of reliability of the provided context information. The inference was performed in less than one millisecond, which proves how quickly the reliability evaluation of our Bayesian network was performed. The result of the evaluation indicates that the provided context information is more than 83% reliable.

The context information reliability evaluation precedes the evaluation of the context source device behavior. As explained previously (Section 4.2.2.2), the proposed system determines the behavior of a context source by computing behavioral indicators, i.e., failed connection rate (ConnRateS_i), failed authentication rate (FARateS_i), and abnormal message size rate (AMSRS_i) sent by the device. ConnRateS_i, FARateS_i, and AMSRS_i are explained in Section 4.3.2.2. Figure 15(a) illustrates a practical case of evaluating the behavior of a context-source device. In this experiment, the following rates of device behavior were collected: ConnRateS_i: 0, FARateS_i: 0, and AMSRS_i: 0. The ConnRateS_i variable is computed as follows: the CS_i failed connection attempts (equal to 0) divided by the total connect packets (equal to 20). The FARateS_i variable is equal to the failed authentication attempts of CS_i (equal to 0) divided by the overall authentication attempts. Finally, the AMSRS_i variable is obtained by dividing the number of messages with abnormal size (equal to 0) by the total number of messages (equal to 50). The computation of these variable values is detailed in Section 4.2.2.2. This demonstrates that this context source did not have any connection or authentication failures and did not send any abnormal messages. The result of the evaluation shows that the device behaves well.

Figure 15(b) illustrates the case of a malicious context source. The algorithm is provided with the following values: ConnRateS_i: 0.5, FARateS_i: 0.4, and AMSRS_i: 0.9. These

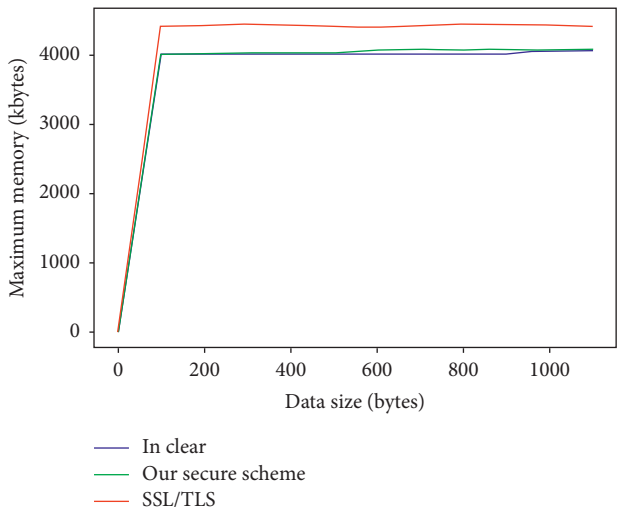


Figure 12: Comparison of systems in terms of maximum memory usage.

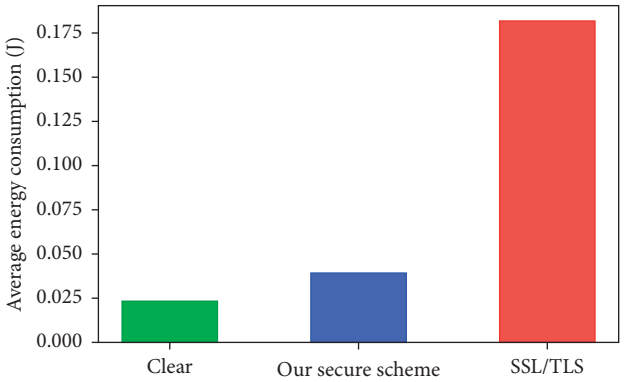


Figure 13: Comparison of systems in terms of energy consumption.

values reflect the malicious activities of this context source. Indeed, these activities can be explained by the following parameters. The failed connection attempts of CS_i are 50 over a total of 100 connection attempts. The failed authentication attempts are 20, and the number of total authentications is 50. The messages with abnormal size detected are 27 over a total of 30 messages. Thus, the result of the behavior evaluation indicates that the context source has a malicious behavior.

The reputation evaluation system uses the outputs from the previous mechanisms to estimate the reputation of the device. This is the last step of the proposed trust management system. Figure 16 shows the obtained results for a doubtful node (Figure 16(b)) and a sure one (Figure 16(a)).

The value of reliability of context information provided by a context source with a good reputation is 90%, and its behavior is evaluated as "good." The context source with a doubtful reputation has a good level of reliability of the context information, evaluated at 80%. Its behavior is rated as "malicious" because it has a high connection failure rate. Thus, our system has a high detection accuracy (equation (17)). It has a false positive rate of less than 4% and an average detection speed of approximately 2 milliseconds. It should also be noted that the detection time is not proportional to the number of context sources. The mechanism has the same detection time with one context source and ten context sources. Indeed, our system performs parallel processing of reputation evaluation. Thus, the proposed mechanism is scalable. Table 5 summarizes the properties of the proposed mechanism on 100 samples [61].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (17)$$

The false negative ratio represents the percentage of unreliable context information detected as reliable. This ratio reduces with enrichment of the user's profile, increasing by the same accuracy of the mechanism. Considering the speed and the efficiency of the detection ensured by our proposed mechanism, we can conclude that our mechanism has a low impact on secure context-awareness management in the aforementioned IoT applications.

5.4. Security and Trust Analysis. The implementation of context-aware security and privacy in the smart city raises several security threats (see Section 2). In this section, we discuss the security, privacy, and trust properties of our proposal.

5.4.1. Security and Privacy. The proposed secure exchange system for context information offers five security services while preserving the user's privacy. First, it limits the DTM access to only authorized devices with mandatory authentication of the latter before any context information is transmitted. Second, with the use of the AES CCM mode, our system ensures the authentication of the data origin by allowing the source to generate an encrypted authentication tag (Section 4.2.4). Only the DTM is able to decrypt with its private key. Third, this mechanism also ensures the integrity of the data exchanged and therefore helps to mitigate attacks on data modification.

Fourth, encrypting context information with AES and using an ephemeral shared secret key guarantee its confidentiality. Thanks to the key exchange integrated into ECIES, only the DTM can decrypt the context information encrypted with its public key and sent by authorized context sources. Thus, the context information is protected against eavesdropping. The proposed identification makes it possible to avoid, in particular, identity spoofing attacks. Fifth, the proposed system guarantees protection against replay attacks by preventing context information that has previously been received from being reprocessed again, even if it comes from trusted context sources. This property is ensured by the built-in antireplay mechanism (see Section 4.2.4). In addition, the association with the reputation management mechanism ensures protection against denial-of-service (DoS) attacks. Indeed, when a device has a suspicious behavior that may be close to a DoS attack, it is quickly detected, and its communications will be rejected. Thus, the proposed system ensures availability.

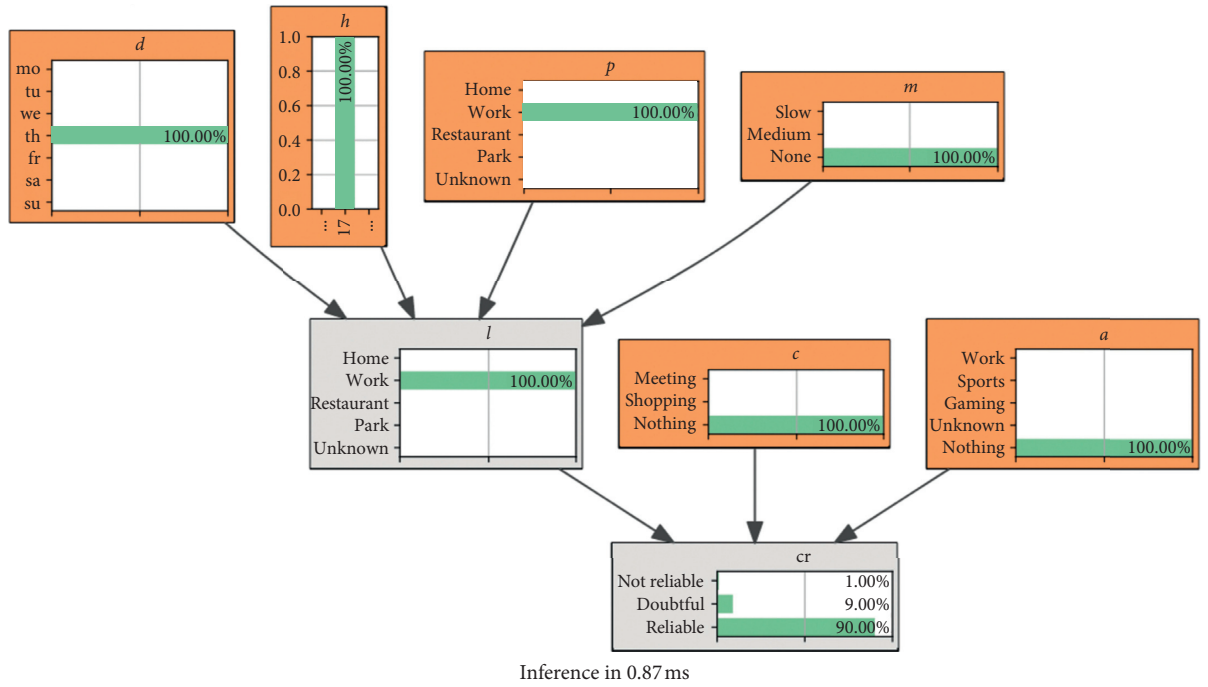


Figure 14: Result of the inference of the Bayesian network for context information evaluation.

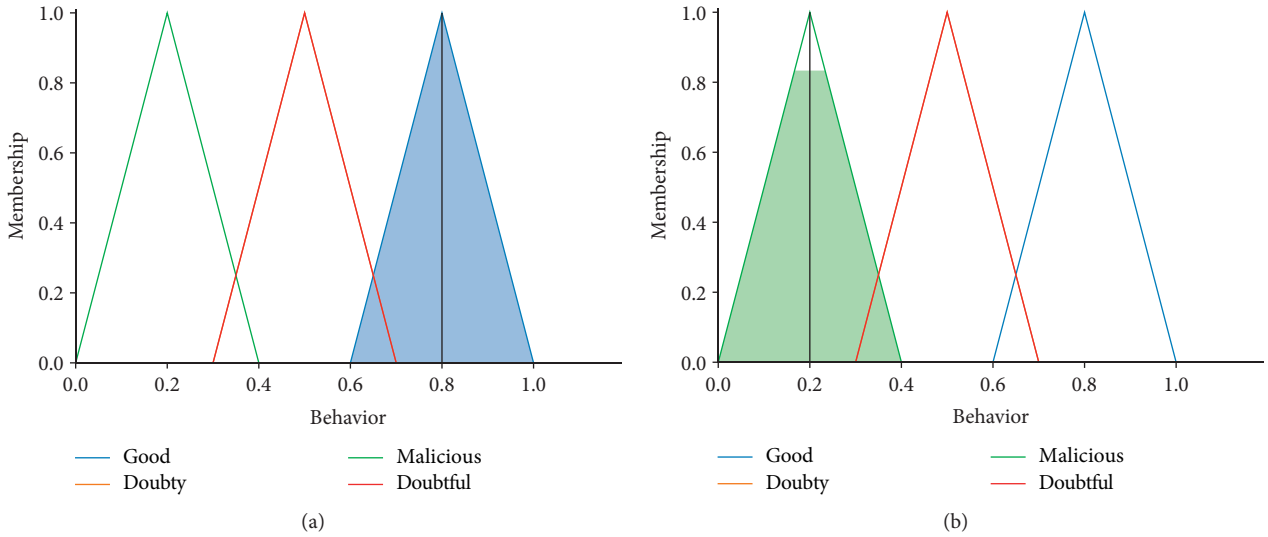


Figure 15: Behavioral evaluation results from two sources of context.

The user's privacy must be preserved in a context-aware security and privacy system because context information largely includes sensitive information about the user (geographical location, current activity, etc.). The proposed system allows the anonymization of context information by ensuring that it is sent without identifying users. By guaranteeing the confidentiality of this information and ensuring that only the DTM can access this information, the proposed system preserves the user's privacy. Another aspect of privacy preservation is the protection of data at the storage level. The proposed system only temporarily stores the data in the encrypted form. It limits the risk of disclosure of the data in case of an attack and thus preserves the user's privacy.

Finally, physical security must be considered. Our system does not store the symmetric encryption keys used by AES. However, it stores the private keys used for more secure exchange operations. Indeed, the ECIES cryptographic system is a hybrid system, i.e., it implements asymmetric and symmetric cryptography. It uses a private/public key pair (Diffie–Hellman exchange). The public key is derived from the private key. The public key is then used by different parties to generate the shared secret key, also called the session key. This secret key is used to encrypt data with a

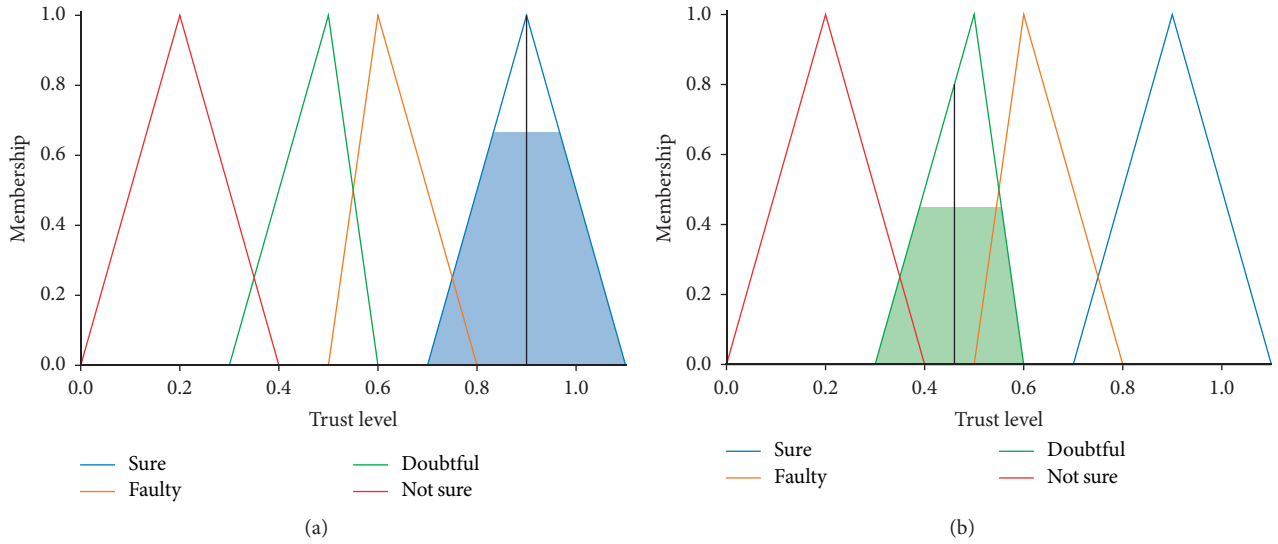


Figure 16: Results of the reputation evaluation from two sources of context.

Table 5: Characteristics of the trust management mechanism.

Properties	Values
Detection time	~ = 2 milliseconds
Accuracy	>81%
True positive (TP)	≥80%
False positive (FP)	<4%
True negative (TN)	≥99%
False negative (FN)	≤19%

symmetric cryptographic system (e.g., AES-CCM). Thus, further work is required to protect the private keys of the devices from key extraction attacks. One possible approach is the use of a Root-of-Trust (RoT) module, composed of a secure element (SE) for secure key storage, a Trusted Platform Module (TPM), or a Trusted Execution Environment (TEE) for key generation and/or derivation operations (public, private, ephemeral secret, etc.) [62].

5.4.2. Trust Management. Generally, a reputation-based trust management system is vulnerable to ballot stuffing and bad-mouthing attacks, identity change attacks, etc. [63, 64] (Section 2). Our proposition allows to mitigate identity change attacks, thanks to identity management integrated in the secure exchange mechanism of context information. It also makes it possible to mitigate attacks on the trust score by evaluating the behavior of context sources and the reliability of the context information they transmit.

In addition, the data used by the Bayesian network to evaluate the reliability of the context information are provided manually. As introduced in Section 3, the UPM module can continuously train the Bayesian network with the new user profile data. Thus, the more the user profile data are, the more effective the Bayesian network will be in determining the reliability of the context information.

6. Conclusion

In this paper, we presented a secure context-awareness management system in an IoT environment. This system ensures the secure exchange of context information and enables the detection of malicious or compromised context sources. It also allows the context-aware security and privacy system, CASPaaS, to avoid making erroneous decisions to adapt security and privacy mechanisms. Indeed, thanks to the antireplay mechanism and the reliability of context information, CASPaaS will process only reliable context information. The proposed solution was implemented and evaluated with MQTT-based communications. The evaluation proved its effectiveness compared to the SSL/TLS protocol. The evaluation also proved the effectiveness of the trust management mechanism in terms of detection accuracy and speed. The overall impact of the solution in a context-aware security and privacy system is acceptable considering the number of nodes a user may have in the smart city. The obtained results show that using the user's profile in detecting unreliable context information can yield good results.

However, this work has some limitations. For example, the used user profile information is static and provided manually. So, it could be interesting to make the user profile information dynamic and automatically enriched. Also, our proposal was validated using data that are provided in lab. So, we have to assess the real effectiveness of our proposal on real-world cases. Thus, we plan to perform tests with real data that we will generate in the near future.

Another interesting future work consists in implementing the solution with hardware security and evaluating its impact on all the mechanisms proposed in this work. In addition, the implementation of some other modules of the proposed CASPaaS architecture [7] is underway, and the achievement of the implementation of the entire system will

allow us to evaluate its overall performance in a practical IoT application such as e-Health.

Data Availability

The experiment results data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] L. You, C. Choi, V. Sharma, I. Woungang, and B. Bhargava, *Guest Editorial:Advances in Security and Privacy Technologies for Forthcoming SmartSystems, Services, Computing, and Networks*, p. 1, AUTOSOFT, London, UK, 2018.

[2] E. de Matos, R. T. Tiburski, L. A. Amaral, and F. Hessel, "Providing context-aware security for IoT environments through context sharing feature," in *Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering*, pp. 1711–1715, TrustCom/BigDataSE), New York, NY, USA, August 2018.

[3] J. L. H. Ramos, J. B. Bernabe, and A. F. Skarmeta, "Managing context information for adaptive security in IoT environments," in *Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pp. 676–681, Gwangiu, South Korea, March 2015.

[4] T. Sylla, M. A. Chalouf, K. Francine, and K. Samake, "Context-aware security in the internet of things: a survey," *IJAACS*, vol. 14, no. 3, p. 1, 2021.

[5] L. Kou, Y. Shi, L. Zhang, D. Liu, and Q. Yang, "A lightweight three-factor user authentication protocol for the information perception of IoT," *Computers, Materials & Continua*, vol. 58, no. 2, pp. 545–565, 2019.

[6] Y. Ashibani, D. Kauling, and Q. H. Mahmoud, "A context-aware authentication service for smart homes," in *Proceedings of the 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 588-589, Las Vegas, NV, USA, January 2017.

[7] T. Sylla, M. A. Chalouf, F. Krief, and K. Samaké, "Towards a context-aware security and privacy as a service in the internet of things," in *Information Security Theory and Practice*, M. Laurent and T. Giannetsos, Eds., pp. 240–252, Springer International Publishing, Cham, Switzerland, 2020.

[8] P. Temdee and R. Prasad, "Security for context-aware applications," in *Context-Aware Communication and Computing: Applications for Smart Environment*, pp. 97–125, Springer International Publishing, Cham, Switzerland, 2018.

[9] J. Zuo, Y. Lu, H. Gao, R. Cao, Z. Guo, and J. Feng, "Comprehensive information security evaluation model based on multi-level decomposition feedback for IoT," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 683–704, 2020.

[10] H. Chen, W. Wan, J. Xia et al., "Task-Attribute-based access control scheme for IoT via blockchain," *Computers, Materials & Continua*, vol. 65, no. 3, pp. 2441–2453, 2020.

[11] B. Le Nguyen, E. Laxmi Lydia, M. Elhoseny et al., "Privacy preserving blockchain technique to achieve secure and reliable sharing of IoT data," *Computers, Materials & Continua*, vol. 65, no. 1, pp. 87–107, 2020.

[12] S. Singh, A. S. M. Sanwar Hosen, and B. Yoon, *Blockchain Security Attacks, Challenges, and Solutions for the Future Distributed IoT Network*, p. 1, IEEE Access, Piscataway, NJ, USA, 2021.

[13] P. N. Mahalle and P. S. Dhotre, "Security issues in context-aware systems," in *Context-Aware Pervasive Systems and Applications*, vol. 169, pp. 137–149, Springer Singapore, Singapore, 2020.

[14] M. Barhamgi, C. Perera, C. Ghedira, and D. Benslimane, "User-centric privacy engineering for the internet of things," 2018, http://arxiv.org/abs/1809.00926.

[15] R. Neisse, G. Steri, G. Baldini, E. Tragos, I. N. Fovino, and M. Botterman, "Dynamic context-aware scalable and trust-based IoT security, privacy framework," in *Internet of Things-From Research and Innovation to Market Deployment*, pp. 199–224, River Publishers, Aalborg, Denmark, 2015.

[16] H. Kashif and L. Wolfgang, "Context-aware authentication for the internet of things," in *Proceedings of the ICAS 2015-The Eleventh International Conference on Autonomic and Autonomous Systems*, Rome, Italy, May 2015.

[17] J. Ahamed and F. Khan, "An enhanced context-aware capability-based access control model for the internet of things in healthcare," in *Proceedings of the 2019 Sixth HCT Information Technology Trends (ITT)*, pp. 126–131, Ras Al Khaimah, UAE, November 2019.

[18] V. Alagar, A. Alsaig, O. Ormandjiva, and K. Wan, "Context-based security and privacy for healthcare IoT," in *Proceedings of the 2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 122–128, Xi'an, China, August 2018.

[19] P. K. Chouhan, S. McClean, and M. Shackleton, "Situation assessment to secure IoT applications," in *Proceedings of the 2018 Fifth International Conference on Internet of Things: Systems, Management and Security*, pp. 70–77, Valencia, Spain, October 2018.

[20] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, and R. Guizzetti, "OSCAR: object security architecture for the internet of things," *Ad Hoc Networks*, vol. 32, pp. 3–16, 2015.

[21] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," 2014.

[22] O. Alphand, M. Amoretti, T. Claeys et al., "IoTChain: a blockchain security architecture for the Internet of Things," in *Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, April 2018.

[23] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, and H. Hashim, "A comparison of lightweight communication protocols in robotic applications," *Procedia Computer Science*, vol. 76, pp. 400–405, 2015.

[24] L. Malina, G. Srivastava, P. Dzurenda, J. Hajny, and R. Fujdiak, "A secure publish/subscribe protocol for internet of things," in *Proceedings of the 14th International Conference on Availability, Reliability and Security - ARES '19*, pp. 1–10, Canterbury, CA, USA, May 2019.

[25] A. Arfaoui, S. Cherkaoui, A. Kribeche, S. M. Senouci, and M. Hamdi, "Context-aware adaptive authentication and authorization in internet of things," in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–6, Shanghai, China, May 2019.

[26] J. Chen, Z. Tian, X. Cui, L. Yin, and X. Wang, "Trust architecture and reputation evaluation for internet of things," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3099–3107, 2019.

[27] R.M Swarna Priya, R. M. Praveen Kumar, M. Parimala et al., "An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture," *Computer Communications*, vol. 160, pp. 139–149, 2020.

[28] S. u. Rehman, M. Khaliq, S. I. Imtiaz et al., "DIDDOS: an approach for detection and identification of distributed denial of service (DDoS) cyberattacks using gated recurrent units (GRU)," *Future Generation Computer Systems*, vol. 118, pp. 453–466, 2021.

[29] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[30] A. Rehman Javed, Z. Jalil, S. Atif Moqurrab, S. Abbas, and X. Liu, "Ensemble Adaboost classifier for accurate and fast detection of botnet attacks in connected vehicles," *Transactions on Emerging Telecommunications Technologies*, 2020.

[31] A. R. Javed, M. O. Beg, M. Asim, T. Baker, and A. H. Al-Bayatti, "AlphaLogger: detecting motion-based side-channel attack using smartphone keystrokes," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[32] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," *Telecommunication Systems*, vol. 76, no. 1, pp. 139–154, 2021.

[33] I.-R. Chen, J. Guo, and F. Bao, "Trust management for SOA-based IoT and its application to service composition," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 482–495, 2016.

[34] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, *MQTT Version 5.0 OASIS Standard*, OASIS Standard, Burlington, MA, USA, 2019, https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

[35] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions," *Journal of Ambient Intelligence and Humanized Computing*, 2017.

[36] M. James and D. S. Kumar, "An implementation of modified lightweight Advanced encryption standard in FPGA," *Procedia Technology*, vol. 25, pp. 582–589, 2016.

[37] J. Daemen and V. Rijmen, "The design of rijndael: AES-the advanced encryption standard," 2002.

[38] R. Housley, "Using AES-CCM and AES-GCM authenticated encryption in the cryptographic message syntax (CMS)," 2007, https://tools.ietf.org/html/rfc5084.

[39] N. Ferguson, "Authentication weaknesses in GCM," 2005, https://csrc.nist.gov/csrc/media/projects/block-cipher-techniques/documents/bcm/comments/cwc-gcm/ferguson2.pdf.

[40] S. Li, L. D. Xu, and I. Romdhani, *Securing the Internet of Things*, Syngress, Cambridge, MA, USA, 2017.

[41] T. Claeys, F. Rousseau, and B. Tourancheau, "Securing complex IoT platforms with token based access control and authenticated key establishment," in *Proceedings of the 2017 International Workshop on Secure Internet of Things (SIoT)*, pp. 1–9, Oslo, Norway, September 2017.

[42] C. Schmitt, M. Noack, and B. Stiller, "TinyTO: two-way authentication for constrained devices in the Internet of Things," in *Internet of Things*, pp. 239–258, Elsevier, Amsterdam, Netherlands, 2016.

[43] H. Gilbert and H. Handschuh, "Security analysis of SHA-256 and sisters," in *Selected Areas in Cryptography*, M. Matsui and R. J. Zuccherato, Eds., vol. 3006, pp. 175–193, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[44] E. Barker, L. Chen, A. Roginsky, A. Vassilev, and R. Davis, *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography*, National Institute of Standards and Technology, NIST Special Publication, Gaithersburg, MD, USA, 2018.

[45] 'R. . Général de Sécurité', "Agence nationale de la sécuritédes systèmes d'information, Mécanismes cryptographiques Annexe B1 Ver.2," 2014, https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B1.pdf.

[46] D. R. L. Brown, "Sec 1: elliptic curve cryptography," 2009.

[47] F. Chen and J. Yuan, "Enhanced key derivation function of HMAC-SHA-256 algorithm in LTE network," in *Proceedings of the 2012 Fourth International Conference on Multimedia Information Networking and Security*, pp. 15–18, Nanjing, China, November 2012.

[48] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *Journal of Network and Computer Applications*, vol. 42, pp. 120–134, 2014.

[49] Z. Yang, K. Zheng, K. Yang, and V. C. M. Leung, "A blockchain-based reputation system for data credibility assessment in vehicular networks," in *Proceedings of the 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–5, Montreal, Canada, October 2017.

[50] N. Brgulja, R. Kusber, K. David, and M. Baumgarten, "Measuring the probability of correctness of contextual information in context aware systems," in *Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, pp. 246–253, Chengdu, China, December 2009.

[51] I. U. Din, M. Guizani, J. J. P. C. Rodrigues, S. Hassan, and V. V. Korotaev, "Machine learning in the Internet of Things: designed techniques for smart cities," *Future Generation Computer Systems*, vol. 100, pp. 826–843, 2019.

[52] J. Pearl, "Bayesian networks," in *The Handbook of Brain Theory and Neural Networks*, pp. 149–153, MIT Press, Cambridge, MA, USA, 1998.

[53] S. Schiaffino and A. Amandi, "Intelligent user profiling," in *Artificial Intelligence an International Perspective*, M. Bramer, Ed., vol. 5640, pp. 193–216, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[54] M. T. Quach, F. Krief, M. A. Chalouf, and H. Khalifé, "Fuzzy-based interference level estimation in cognitive radio networks," 2014.

[55] "Secure-MQTT: an efficient fuzzy logic-based approach to detect DoS attack in MQTT protocol for internet of things," *Journal of Wireless Networking and Communications*, vol. 2019, no. 1, 90 pages, 2019.

[56] Eclipse Foundation, *eclipse/paho.mqtt.embedded-c*, Eclipse Foundation, Nepean, Canada, 2020.

[57] W. Dai, "Crypto++ library 8.2 | free C++ class library of cryptographic schemes," 2019, https://www.cryptopp.com/.

[58] 'Eclipse Mosquitto', Eclipse Mosquitto, 2018. https://mosquitto.org/.

[59] aGrUM team, "aGrUM/pyAgrum," 2020, https://agrum.gitlab.io/.

[60] 'SciKit-Fuzzy — skfuzzy v0.2 docs'. 2020, https://pythonhosted.org/scikit-fuzzy/overview.html.

[61] A. Baratloo, M. Hosseini, A. Negida, and G. El Ashal, "Part 1: simple definition and calculation of accuracy, sensitivity and specificity," *Emerg (Tehran)*, vol. 3, no. 2, pp. 48-49, 2015.

[62] SCA, *Embedded Hardware Security for IoT Applications*, Smart Card Alliance, Princeton Junction, NJ, USA, 2016, https://www.securetechalliance.org/wp-content/uploads/Embedded-HW-Security-for-IoT-WP-FINAL-December-2016.pdf.

[63] J. M. Such, "Attacks and vulnerabilities of trust and reputation models," in *Agreement Technologies*, S. Ossowski, Ed., pp. 467–477, Springer Netherlands, Dordrecht, Netherlands, 2013.

[64] D. Fraga, Z. Bankovic, and J. M. Moya, "A taxonomy of trust and reputation system attacks," in *Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 41–50, Liverpool, UK, June 2012.