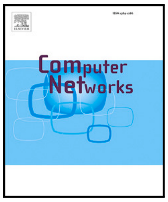
PARFAIT

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



PARFAIT: Privacy-preserving, secure, and low-delay service access in fog-enabled IoT ecosystems

Savio Sciancalepore

Eindhoven University of Technology, Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
IoT
Fog computing
Edge computing
Security
Privacy

ABSTRACT

Traditional fog-enabled IoT ecosystems always assume fully-trusted and secure fog nodes, offering computational capabilities and storage space closer to constrained IoT devices. However, such security-related assumptions can easily fall when considering the exposure of fog nodes' location, the heterogeneity of device providers, and the ease of misuse and misconfigurations by end-users, to name a few. As a result, compromised fog nodes can stealthily steal sensitive information, such as the devices' location, path, and private personal attributes.

This paper presents PARFAIT, a privacy-preserving, secure, and low-delay framework for securely accessing services in fog-enabled IoT ecosystems. PARFAITguarantees low-delay authentication and authorization to local fog nodes, protecting the identity and the attributes possessed by the IoT devices. Moreover, PARFAITuses rolling ephemeral identities, providing unlinkability among access requests, thus preventing the tracking of mobile IoT devices by multiple compromised fog nodes. We performed several experimental tests on a reference proof-of-concept to show the viability of PARFAIT. Specifically, adopting an elliptic curve with a group size of 512 bits, PARFAITallows the access to a single protected resource in only 0.274 s, and such a delay rises to only 0.359 s with 10 consecutive requests (66.8% less than the quickest competing approach).

1. Introduction

The *fog computing* paradigm is nowadays gaining momentum in both Academia and Industry, thanks to the enormous benefits it can bring in many application scenarios based on the Internet of Things (IoT) [1–3].

However, deploying fog nodes in IoT ecosystems also brings new security challenges [4]. Indeed, differently from Cloud-enabled IoT scenarios, the positioning of critical processing units in fog-enabled IoT ecosystems is known and exposed to adversarial entities [5]. Even though security (both hardware and software) is one of the pillars enabling the fog computing paradigm [6], the involvement of a plethora of different manufacturers, possible misconfigurations due to end-users inexperience, and the deployment of the fog devices in the wild, all make fog nodes an ideal target of cyber-attacks [7,8].

Due to the sensitive location of their deployment, an adversary compromising one or multiple fog nodes To the best of our knowledge, the existence of multiple compromised fog nodes in fog-enabled IoT ecosystems has not been considered in the literature, yet. Indeed, the vast majority of scientific contributions assumed *trusted* fog nodes, used to outsource computationally intensive and energy-demanding tasks. At the same time, the very few works considering untrusted fog devices always considered selfish fog nodes, not colluding with additional network elements to gain information about IoT devices (see Section 2 for a comprehensive overview).

Contribution. This paper proposes PARFAIT (an acronym for PrivAcy-pReserving Fog-enAbled IoT), a privacy-preserving, secure, and low-delay framework for securely accessing services in fog-enabled IoT ecosystems. PARFAIT enables Cloud-less authentication and authorization in fog-enabled IoT scenarios, while at the same time protecting IoT devices' privacy (identity, location, path, and private personal attributes). In summary, PARFAIT turns an encryption scheme, i.e., the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) technique, typically used statically within traditional IT systems, into an interactive challenge–response blind authentication and authorization scheme. Moreover, cryptographic operations leveraging costly bilinear pairings are used only once in PARFAIT, to demonstrate the possession of a suitable set of access rights, rather than at every service access. PARFAIT also does not necessarily rely on the Cloud for authenticating and authorizing IoT devices to fog nodes, significantly reducing delays.

A proof-of-concept of PARFAIT has been implemented in Python, tested, and cross-compared against competing solutions. The comparison reported in Section 7 shows that, configured with an elliptic curve group of size of 512 bits, PARFAIT allows to use a service securely in

E-mail address: s.sciancalepore@tue.nl.

https://doi.org/10.1016/j.comnet.2022.108799
Received 15 July 2021; Received in revised form 18 October 2021; Accepted 17 January 2022
Available online 29 January 2022

only 0.274 s. In comparison, the access delay increases to only 0.359 s with 10 consecutive requests, which is approximately 66.8% less than the quickest available competing approach. Overall, this paper provides the following contributions:

- shedding lights on the significant privacy issues that can arise when delegating authentication and authorization functionalities to fog nodes in IoT ecosystems;
- designing PARFAIT, a low-delay, privacy-preserving, and secure framework for resources access in IoT ecosystems adhering to the fog computing architectural paradigm;
- turning the CP-ABE encryption algorithm into an interactive access control scheme, based on the JWT tool and an anonymous challenge–response mechanism;
- integrating the CP-ABE scheme into fog-enabled IoT ecosystems, without sacrificing the location and identity privacy of IoT nodes;
- providing a low-delay security scheme, outperforming competing solutions in terms of resource access delay and security functionalities.

Roadmap. The rest of this paper is organized as follows. Section 2 illustrates the related work, Section 3 provides the background on the main building blocks of the solution, Section 4 introduces the considered scenario and the adversary model, Section 5 provides the details of PARFAIT, Section 6 focuses on the security of PARFAIT, Section 7 describes the details of the implementation and performance assessment, and finally, Section 8 draws the conclusions.

2. Related work

Many contributions in the last years focused on authentication and authorization issues in fog-enabled IoT ecosystems.

The authors in [9] focused on low-latency authentication, and avoided authentication requests to pass from the core cellular network, by decentralizing the related procedures to the fog layer. To this aim, they designed two solutions that achieve low-latency authentication at the fog layer, while guaranteeing a transparent integration in the existing network architecture. However, their solution did not address access control issues, and it does not address the security issues arising from multiple colluding fog nodes.

Protecting the privacy of the users and IoT devices while moving in fog-enabled IoT ecosystems has been mainly considered in vehicular networks. For instance, the authors in [10] proposed a Cloud-assisted mutual authentication process. The protocol proposed by the authors is lightweight, as it does not require cryptography operations on the moving devices. However, the protocol exposes the vehicles' identity to the fog nodes, and thus they can track their movements by colluding and sharing information. In addition, no considerations about access control are provided.

The authors in [11] proposed an identity-based anonymous authenticated key agreement protocol for Mobile Edge Computing (MEC) scenarios. Using a single authentication provided by a CSP, the protocol allows IoT devices to authenticate with many fog nodes, guaranteeing untraceability and perfect forward secrecy properties, to name a few. Unfortunately, the authors considered an adversary model that did not take into account colluding fog nodes but considered fog nodes selfish. When many fog nodes collude, the assumed security properties fall, allowing fog nodes to track user movements. In addition, access control issues are not considered.

The authors in [12] proposed a Software-Defined Networking (SDN)-based handover authentication scheme suitable for MEC scenarios. The authors considered Edge devices as constrained as IoT devices, and integrated the SDN network architecture into the core cellular network design. Despite their authentication protocol being quicker than traditional fully Cloud-based solutions, the protocol still requires an interaction between the Edge node and the Cloud, making the whole procedure still dependent on the Cloud. In addition, authorization issues are not considered, and colluding Edge/fog nodes are not taken into account within the adversarial model.

Many contributions in the last years focused on the combined key agreement–authentication problem in fog computing systems. For instance, the authors in [13] designed SAKA-FC, an authentication and key agreement scheme using only lightweight operations, such as hashing functions and eXclusive-OR (XOR) operations. Similarly, the authors in [14] proposed three lightweight authentication protocols, achieving mutual low-latency authentication, key exchange, and protection against external eavesdroppers. However, the authors' scenario and the security objectives assumed in these contributions are different from this manuscript. First, these contributions assume that the IoT devices cannot support any cryptography operations. In contrast, the present contribution considers state-of-art devices, that can execute cryptography operations, also through low-cost hardware support. In addition, these contributions consider neither colluding fog nodes nor the capability of the fog devices to track the movements of the IoT devices. Finally, authorization issues are not considered, as well.

Several recent contributions used the CP-ABE scheme and its variants combined with the fog Computing architecture. To name a few, the authors in [15] used the fog nodes to outsource computations that cannot be carried out by constrained IoT devices, decorrelating their computational burden from the number of involved attributes. In addition, they also propose a method to update the attributes efficiently. However, outsourcing encryption and decryption operations to untrusted fog nodes generates severe privacy issues, exposing the identity, attributes, and location of the involved IoT devices. Similar problems can also be found in [16], where the authors proposed a privacy-preserving access control scheme assisted by fog nodes. Although the authors modified the legacy CP-ABE scheme to provide anonymous authentication, the protocol still requires continuous interaction with the Cloud, revealing all the users' attributes to untrusted fog nodes. Similar issues also apply for the contribution in [17], where the authors outsource computations to fog devices. Similar issues apply also to very recent works, such as [18], where the authors proposed an approach based on Multi-Authority Attribute-Based Encryption (MA-ABE) supporting revocation and outsourcing attributes computation to the fog. At the same time, the authors in [19,20] presented a data sharing system including bilateral access control based on lightweight matchmaking encryption, outsourcing costly operations to fog nodes. Despite the fog nodes are conceived as semi-trusted entities, such schemes only ensure the correctness of operations and the privacy of the attributes, while they cannot protect IoT sensors' location privacy, especially when multiple colluding fog nodes are considered.

The authors in [21] proposed a protocol based on CP-ABE, achieving encrypted key exchange, fine-grained authorization, confidentiality, and authentication. However, they considered a different scenario than this contribution, where two fog nodes need to authenticate, establish a secure connection, and access authorized data. Therefore, the Cloud is continuously involved in the scenario, and the privacy IoT devices is neglected. Finally, mobility and unlinkability of the IoT devices are not considered.

It is also worth mentioning the recent contribution by authors in [22], providing a solution to use a state-less and potentially untrusted device to perform cryptographic operations such as signature validation and encryption using computationally-constrained IoT devices. On the one hand, the architecture of such a solution is compatible with the one proposed in this manuscript, representing a viable solution. On the other hand, such a solution has been proposed and tested only with standard public-key crypto-systems (RSA and ECC). As we will show in Section 4.3, the need to protect the privacy of the attributes of the IoT devices in fog-enabled IoT ecosystems requires the usage of solutions such as CP-ABE, which operate differently than traditional public-key crypto-systems. Thus, how to adapt the proposal in [22] to work with pairing-based strategies such as CP-ABE is unclear, and require additional dedicated studies.

The authors in [23] proposed ADVOCATE, a framework that facilitates the processing of personal data in IoT environments via processes compliant with the latest GDPR rules in the EU context. On the one hand, the topic tackled by this work is different than the one tackled in our manuscript, as the cited work provides a method to control the usage of shared data, while we try to minimize disclosed data. On the other hand, our work is integrable with the cited work to provide a method for users and IoT device to maintain full control over attributes and other personal data.

Similarly, The authors in [24] proposed to apply Moving Target Defense (MTD) techniques to protect devices' identity during data transmission. However, the authors only look at anonymity, without considering other private data such as location and attributes, that can easily lead to users identification.

Finally, note that several works that recently investigated handover procedures in mobile fog-enabled IoT networks, such as [25–27], and [28], to name a few. However, these schemes mainly look at the performance of the communication link during the movement of the IoT devices. At the same time, they do not discuss authentication and authorization functionalities during the connection to different fog nodes.

Compared to the valuable contributions discussed above, PARFAIT integrates the CP-ABE cryptography technique and the JSON Web Token (JWT) tool into fog-enabled IoT ecosystems in a novel way, such that the related computational and bandwidth requirements could be mitigated. A critical novel element distinguishing PARFAIT from the current state of the art is that the CP-ABE technique is not used to provide data confidentiality. Still, it is smartly integrated into an interactive challenge–response authentication and access control scheme. As a result, IoT devices (and particularly the IoT gateway) execute CP-ABE decryption operations only once, during the connection establishment with the fog node hosting the requested resources. The innovative usage of CP-ABE and JWT in PARFAIT also allows providing enhanced security compared to the previously-published approaches, especially when fog nodes may be partially untrusted. Indeed, PARFAIT do not expose the attributes and the identity of the IoT sensors to potentially compromised fog nodes, emerging as a privacy-preserving solution even in the presence of the adversary described in Section 4.2.

Section 7.3 will provide a detailed comparison of the solution presented hereby with the discussed related work, along reference system and security requirements.

3. Preliminaries

This section introduces the readers to the main building blocks of PARFAIT. Specifically, Section 3.1 introduces the JSON Web Token (JWT) tool, while Section 3.2 provides background information on the CP-ABE crypto-system.

3.1. JSON web tokens

JSON Web Tokens are a security tool introduced in [29] and standardized by IETF. Specifically, JWTs are a digitally-signed structure, that uniquely binds some information (namely, *claims*) to the creator of the token, so that the resulting object can be easily transferred among a set of parties.

JWTs include many *standardized claims*, i.e., information fields whose meaning and definition have been defined globally. A few examples include the *issuer* claim (iss), uniquely identifying the entity that created the JWT, the *subject* (sub), containing the information about the entity for which the JWT has been created, the *timestamp* (iat), that indicates the creation time of the JWT, and the *expiration date* (exp), specifying the maximum validity time of the JWT. In addition, JWTs allow for the creation of *additional claims*, whose meaning can be specified by the system administrator.

At the end of the JWT, there is a *sign* field (sgn), that binds together all the information. It is generated by concatenating all the claims in the structure, hashing the resulting string with a hashing function, and signing the resulting hash value using the private key of the *issuer*.

Therefore, at the reception of a JWT from a remote entity, provided that the recipient trusts the issuer of the JWT, by verifying the signature of the received structure the recipient can be sure that all the information contained in the JWT are authentic and not tampered.

This work extends the traditional structure of the JWT with additional claims, such as the *ephemeral attribute* claim (eat), containing information about the time-limited attribute released by the CSP.

3.2. Ciphertext-Policy Attribute-Based Encryption

The CP-ABE crypto-system has been introduced in 2007 by the authors in [30]. The idea at the roots of this scheme is to remove the presence of a trusted server in the management of access control procedures, and to provide a cryptography mechanism that allows any entity to evaluate the possession of a set of attributes declared by a user.

Overall, a generic CP-ABE crypto-system consists of five procedures.

- *Setup.* This phase is meant to initialize the public parameters of the system. It takes as input a security parameter λ, and it provides as output a set of public parameters PK and a master key MK.
- *Encrypt.* This process is dedicated to the encryption of a clear-text message. It takes as input a cleartext M, the public parameters PK, and an access structure $\mathcal{A}$ defined over the set of attributes $\mathcal{S}$ defined in the system. It provides as an output a ciphertext C.
- *Key Generation.* this phase aims to equip a user with a key that embeds the set of attributes in its possession. It takes as input the master key MK and the set of attributes $\mathcal{S}_n$ in possession of the nth user, and it provides as an output a key SK_n.
- *Delegate.* Using this procedure, a user can delegate part or all of the attributes in its possession to another user. This function takes as input the key SK_n assigned to the nth user and an access structure $\tilde{\mathcal{S}}$, with $\tilde{\mathcal{S}} \subseteq \mathcal{S}$, and it outputs a new secret key SK_m, with $m \neq n$.
- *Decrypt.* This phase allows a user to decrypt a ciphertext, only if the key in its possession has been generated using a set of attributes used for the generation of the ciphertext. Specifically, it takes as input the ciphertext C, the key SK_n of the user n, and the public parameters PK, and it can return either the cleartext message M or an error $\perp$.

Note that today several amendments of the legacy CP-ABE crypto-system are available, each optimizing a specific feature. For instance, the authors in [31] provided constant-size ciphertexts, the authors in [32] focused on the support of advanced access structures, while the authors in [33] provided optimizations tailored for broadcast communications. This paper adopts the legacy scheme introduced in [30]. However, the scheme can be further generalized and formulated in conjunction with any of the above crypto-systems, provided that they do not require any further network element.

Finally, note that this paper does not simply integrate blindly the scheme proposed in [30]. Conversely, the proposed protocol turns the original off-line encryption scheme into an interactive real-time authentication and authorization protocol. In addition, the proposed scheme provides additional time-limited and privacy-preserving authentication and authorization features, that were not included in the legacy scheme, thanks to the integration with the JWT structure. Finally, the proposed scheme integrates the CP-ABE crypto-system in a network architecture tailored for mobile fog-enabled IoT ecosystems, providing a practical solution to achieve low-latency Cloud-less authentication and authorization procedures.

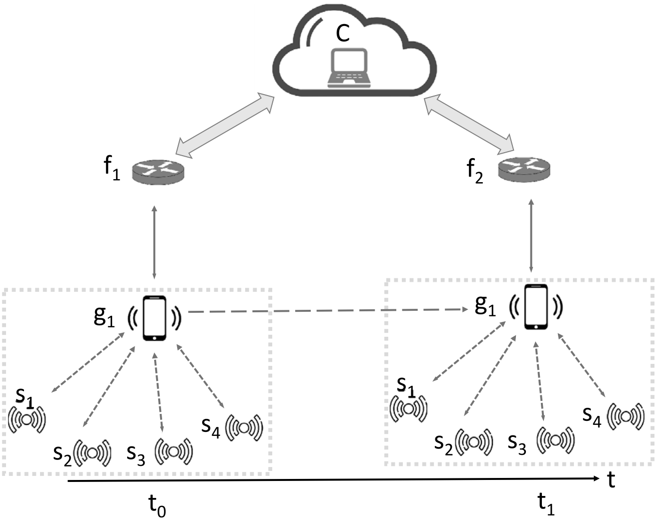


Fig. 1. Reference scenario.

4. Scenario, adversary model, and system requirements

This section presents the system model, the adversarial model, and the system security requirements assumed in this work. Specifically, Section 4.1 introduces the scenario, Section 4.2 details the features and the goal of the adversary, while Section 4.3 introduces and motivates the system security requirements addressed by this work.

4.1. System model

The scenario assumed in this work is depicted in Fig. 1.

In line with many contributions working on the application of the fog computing architecture in the IoT [3,5], this manuscript assumes a network model based on four layers. At the layer-1, there are the *Constrained Devices*, i.e., s_n, with $n \in [1, N]$, referred to as *IoT sensors*. In line with common assumptions and the IETF terminology, IoT sensors interact with the surrounding environment, by sensing physical measurements and modifying the state of surroundings through dedicated sensors and actuators [34]. They are also equipped with a tiny processing unit, that allows them to execute processing tasks. In line with many contributions in the literature [35], this paper assumes that IoT sensors have enough computation, storage, and energy resources to execute symmetric and asymmetric cryptography operations. Thus, they can execute lightweight cryptography protocols and secure wireless connections, e.g., via symmetric encryption keys [36].

The IoT sensors can deliver and receive messages via standard wireless connections, e.g., based on the widespread IEEE 802.15.4 or Bluetooth communication technologies, and they communicate directly with the *gateway* of the IoT network, referred to as the *IoT Gateway*. The IoT gateway, located at the layer-2 of the architecture, can be a mobile phone or a dedicated device, such as the ones used for eHealth applications. As such, the IoT gateway is generally more powerful and less energy-constrained than the underlying IoT sensors. The IoT gateway, namely g_1 in Fig. 1, collects all the traffic directed to/from the IoT network, and proxies it to/from the upper-layer. Specifically, it is assumed that the IoT gateway is equipped with a communication module, that allows it to connect wirelessly to the upper-layer. The gateway manages authentication and authorization procedures with the upper-layer, on behalf of constrained devices.

Note that the gateway and the IoT sensors do not need to be physically separated. Indeed, they could be also installed in the same device and run as separate processes within the same IoT device, in line with many typical IoT devices (note the light gray box in Fig. 1) [37]. We also remark that the function of an IoT gateway is to connect the local IoT network to the external network, i.e., to a network different from the IoT one. Such external network can be either the public Internet or another local network, i.e., the Local Area Network (LAN) where the IoT sensors are connected. In the first case, the devices feature a Subscriber Identity Module (SIM) card, that allows connecting to the public Internet via a specific subscription plan. In the second case, the devices features a Wi-Fi interface, that allows interacting with devices in the LAN as regular Wi-Fi nodes. This latter architecture is adopted by the vast majority of IoT devices, including Smart Home, Smart City, and Healthcare ones. Overall, our target network architecture specifically considers all the use-cases where the resources generated by the IoT sensors are made available to users only through a LAN connection, even when SIM cards are available.

Overall, independently from being integrated on a single device or distributed across multiple sensors, the IoT sensors and the gateway constitute the *Trusted Domain* of the user and, in line with common assumptions, they are assumed to be honest and secure.

This paper also assumes that the IoT gateway and the IoT sensors are mobile; thus, they move across several locations in short time frames. Note that moving the IoT sensors and the gateway together does not raise any mobility issues, as these systems are portable in nature and characterized by a reduced form factor.

The IoT gateway interacts with the *fog nodes* f_q, with $q \in [1, Q]$. In line with common assumptions, the role of the fog nodes is to reduce the delay in accessing the services traditionally hosted on the CSP. Indeed, *fog nodes* hosting useful resources are deployed closer to the IoT sensors, in a way to mitigate the high variability of the data reporting delays and data access latencies characterizing Cloud-based services. As such, fog nodes have to verify the authenticity of the IoT gateway, and to provide the requested services, based on the specific access rights. For instance, they can allow data upload, and data storage, to name a few.

Finally, the layer-4 of the architecture is the CSP C, including all the services traditionally included in the core network. To name a few, they include the *Authentication Center (AuC)* of the network, responsible for user/device authentication, and other processes responsible for the identification of the services each device can access to, based on the related subscription plan.

In the following, we assume that the requests for resources hosted on fog nodes are initiated by the IoT sensors, and routed to the fog nodes through the IoT gateway. However, the protocol flow illustrated in Section 5 does not change even if the requests are performed from outside of the local network, e.g., from users connected to the Cloud.

The system architecture assumes that the communication between the IoT sensors and the IoT gateway is secured through well-known symmetric encryption algorithms, such as Advanced Encryption Standard (AES). The key used to encrypt and decrypt packets could be set up in many ways, e.g., it can be pre-shared or dynamically acquired, e.g., via Elliptic Curve Diffie-Hellman (ECDH)-based protocols [35]. In addition, the communication link between the IoT gateway and the fog node is secured via the well-known Secure Socket Layer (SSL)/Transport Layer Security (TLS) protocol. Note that these assumptions are fully in line with the capabilities of the involved devices, as previously introduced. In addition, they do not impact on the novelty of this contribution, as the aim of this work is to guarantee low-latency and privacy-preserving authentication, authorization, and unlinkability of the requests coming from the IoT network, in the presence of the adversary described in Section 4.2. The notations used in the rest of the manuscript are summarized in Table 1. Note that we used boldface letters (e.g., $\mathbf{A}$) to denote vectors.

Table 1
Notation used throughout the paper.

Notation	Description
N	Overall number of IoT sensors in possession of the end-user.
K	Overall number of IoT gateways.
Q	Overall number of fog nodes.
C	Cloud Service Provider.
s_n	Generic nth IoT sensor.
g_k	Generic kth IoT gateway.
f_q	Generic jth fog node.
$\mathcal{A}$	Generic adversary.
T	Validity time of the cryptography materials released by C.
H	Generic hashing function.
$\mathcal{G}_0$, $\mathcal{G}_1$	Bilinear groups.
p	Order of the bilinear group.
γ	Generator of the bilinear group.
α, β	Nonces extracted by the AzA.
U_n	Username of the IoT sensor s_n.
pwd_n	Password of the IoT sensor s_n.
$info_n$	Additional information of the IoT sensor s_n.
Z	Number of attributes assigned to the IoT sensor s_n.
$\mathbf{A_n}$	Set of attributes assigned to the IoT sensor s_n.
$a_{n,z}$	Generic z-th attribute assigned to the IoT sensor s_n.
P_{AtA}	Public key of the Authentication Authority.
p_{AtA}	Private key of the Authentication Authority.
C_{AtA}	Public key Certificate of the Authentication Authority.
J	Ephemeral identities and attributes assigned to an IoT sensor.
$\epsilon_\mathbf{n}$	Set of ephemeral identities assigned to the IoT sensor s_n.
$\epsilon_{n,j}$	Generic ephemeral identity assigned to the IoT sensor s_n.
$\chi_\mathbf{n}$	Set of ephemeral attributes assigned to the IoT sensor s_n.
$\chi_{n,j}$	Generic ephemeral attribute assigned to the IoT sensor s_n.
$\mathbf{SK_n}$	Set of ephemeral secret keys assigned to the IoT sensor s_n.
$SK_{n,j}$	Generic ephemeral secret key assigned to the IoT sensor s_n.
A_n^+	Temporary vectors for the creation of the ephemeral secret keys.
r, r_z	Nonces used for the creation of the ephemeral secret keys.
$\tau_\mathbf{n}$	Set of ephemeral tokens assigned to the IoT sensor s_n.
$\tau_{n,j}$	Generic ephemeral token assigned to the IoT sensor s_n.
$sign_{n,j}$	Signature of the ephemeral token j assigned to the IoT sensor s_n.
$ts_{n,j}$	Creation time of the eph. token j assigned to the IoT sensor s_n.
l_n	Generic resource requested by the IoT sensor s_n.
ρ_l	Generic access policy assigned to a resource l.
$\rho'_{l,n,j}$	Ephemeral policy for verifying the possession of the attributes in $\tau_{n,j}$.
$c_{l,n,j}$	Random value extracted by the fog node.
$\sigma_{l,n,j}$	Challenge generated by the fog node.
$\varphi_{l,n,j}$	Response computed by the gateway for the IoT sensor s_n.

4.2. Adversary model

The adversary assumed in this work, namely $\mathcal{A}$, features both passive and active capabilities. First, $\mathcal{A}$ is a global passive eavesdropper on the layer-1, layer-2, and layer-3 of the network architecture discussed in Section 4.1. Indeed, thanks to dedicated wireless interfaces, $\mathcal{A}$ can detect and decode (but not decrypt) the content of all the messages exchanged between the IoT sensors and the gateway, as well as the communications between the IoT gateway and the fog nodes.

Moreover, $\mathcal{A}$ can collude with one or more fog nodes, and try to infer on the identity and movement patterns of the IoT sensors by combining the knowledge acquired on the channel with the information gained through compromised fog nodes. Specifically, the fog nodes follow an *honest-but-curious* model, where the compromised fog node behaves honestly in the network, but silently tries to access the information delivered by the IoT sensors and use them for additional objectives, not explicitly authorized by the user.

The active capabilities of $\mathcal{A}$ are used when compromising the fog node. Note that the specific method used by the adversary to compromise the fog node is out of the scope of our manuscript. Indeed, $\mathcal{A}$ can use many possible active attack strategies, e.g., malware infection of social engineering techniques. Therefore, the behaviors described hereby only take into account the operation of $\mathcal{A}$ after compromising the fog node.

Furthermore, $\mathcal{A}$ can also transmit messages on the wireless communication channels, by either replaying previously-exchanged packets or injecting forged messages. To this aim, $\mathcal{A}$ can use rogue IoT sensors or rogue IoT gateways, set up ad-hoc to achieve the attack. Finally, $\mathcal{A}$ could deploy *rogue fog nodes*, and register them to the CSP, to improve its chances to access sensitive information.

Overall, the goals of the adversary are manifold. First, $\mathcal{A}$ would like to stealthily obtain information necessary to authenticate on behalf of a legitimate IoT sensor to the network. Second, $\mathcal{A}$ is willing to obtain information on the access rights in possession of the impersonated IoT sensor(s), so that to possibly operate on its behalf and to access services offered by other (legitimate) fog nodes. Finally, through the collaboration with compromised fog nodes, $\mathcal{A}$ aims to track legitimate IoT sensors, to infer private information about their resource requests and moving patterns.

An example of the adversary model described above is a *Smart City* scenario, where multiple fog nodes are deployed in different areas of the city to optimize the response time and service delay to the users. In particular, when an IoT device of a user requests a service, the closest fog node in the area replies to the request, instead of routing such request to the Cloud over the Internet. Such fog node may operate regularly (as intended) on the network, but then also share information to track the users in their movements, attempting at their privacy. Such information include not only their identity, but also the attributes and cryptography materials provided by the users when accessing services, that may easily lead to their identification.

Protection against additional attacks, such as Denial of Service (DoS), requires the setup of dedicated protection techniques, and thus, they are not in the scope of our manuscript.

Recall that the fog nodes are honest-but-curious: thus, they follow the correct execution of the protocol, and they do not actively interfere with the described operations (e.g., node authentication).

4.3. System and security requirements

In the context of the network architecture discussed above, considering the capabilities and the goals of the adversary introduced in Section 4.2, a security protocol tailored for the described scenario should be able to fulfill at least the system and security requirements listed below.

- **Cloud-Less Authentication.** The system should access authentication services hosted by the Cloud (layer-4) only once in a specific time-frame T. For all the authentication requests occurring in a time $t < T$, the system should be able to authenticate the IoT sensor without accessing Cloud-based services.
- **Cloud-Less Authorization.** The system should access authorization services hosted by the Cloud (layer-4) of only once in a specific time-frame T. For all the authorization requests occurring in a time $t < T$, the system should be able to verify the access rights of the end-user/IoT sensors without accessing Cloud-based services.
- **IoT sensor(s) Identity Privacy.** The identity of the IoT sensors and the IoT gateway accessing the network through the fog node(s) should be protected, to avoid information leakages to untrusted fog nodes.
- **IoT sensor(s) Attributes Privacy.** The attributes of the IoT sensors and the IoT gateway accessing the network through the fog nodes should be protected, to avoid information leakages to untrusted network devices.
- **Device(s) Location Privacy.** The location and the movement patterns of the IoT sensors and the IoT gateway accessing the network through the fog nodes should be protected, to avoid tracking of the end-user bringing the devices and leakage of sensitive location information.

To the best of our knowledge, considering the adversary model described in Section 4.2, the solution described in the following sections is the first that can achieve all the requirements listed above, at the same time.

5. The PARFAIT framework

This section illustrates the details of PARFAIT. Specifically, Section 5.1 introduces the main entities involved in the scenario, while Section 5.2 describes all the phases of PARFAIT.

5.1. Actors

The system model considered hereby involves the following actors.

- *Authentication Authority (AtA)*. It is a network element placed in the Cloud (layer-4), and it is responsible for the authentication of the IoT gateway and the IoT sensors requesting access to the system. It also stores the credentials of the IoT sensors, used to verify the login information and provide access to the network services.
- *Authorization Authority (AzA)*. It is a network element placed in the Cloud (layer-4), and it is responsible for managing access control in the system. Specifically, it releases CP-ABE secret keys, that can be used to prove the possession of a set of attributes satisfying the access policy of a given resource.
- *fog Node(s)*. They are deployed closer to the IoT sensors, e.g., within the access point provided by the network operator and installed in each apartment or industry. They host some services, e.g. computationally-intensive and storage-demanding applications, so that they can be accessed without requiring interactions with the Cloud. Before providing access to the service, the fog Node(s) have to verify that the requesting entity is authenticated, and it has a suitable and fresh set of access rights, required for accessing the requested service. Note that the strong assumption made in this work is that, due to their deployment location, fog nodes cannot be fully trusted. Specifically, in line with the adversarial model described in Section 4.2, fog nodes are *honest-but-curious*, i.e., they follow the rules of the protocol but, at the same time, they would like to obtain private information about the system and users.
- *IoT Gateway*. This network element provides the IoT sensor(s) with external Internet access. It gathers data from the IoT sensor(s) and forwards the traffic to the fog Node(s). In addition, if the IoT sensor is computationally-limited, it also manages authentication and authorization procedures on behalf of the single IoT sensors.
- *IoT sensor(s)*. They interact with the surrounding environment, by either performing sensing activities or actuation tasks. They can communicate with each other, but any communication intended to reach entities outside of the IoT network has to pass through the IoT Gateway.

The following sections describe PARFAIT, a solution that allows the IoT sensors to perform accelerated Cloud-less privacy-preserving authentication and authorization procedures with multiple fog nodes, without involving multiple Cloud-based interactions.

5.2. Protocol details

The PARFAIT protocol is mainly divided into four phases, namely, the *Setup Phase*, the *Registration Phase*, the *Cloud Authentication Phase*, and the *Fog Authentication and Authorization Phase*.

Setup Phase. The aim of this phase is to initialize the system cryptography parameters. It is executed only once on the AzA, at the boot-up of the system. As depicted in Fig. 2, it consists of the following steps.

- The AzA selects two bilinear groups $\mathcal{G}_0$ and $\mathcal{G}_1$ of order p, and a generator γ of this group.

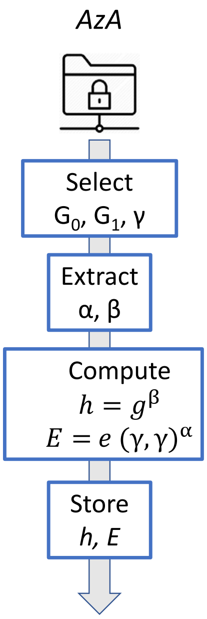


Fig. 2. Sequence diagram of the *Setup Phase* of PARFAIT.

- Then, the AzA extracts two random values, namely α and β, with $\alpha, \beta \in \mathcal{Z}_p$. Starting from these values, it computes the parameters $h = \gamma^\beta$ and $E = e(\gamma,\gamma)^\alpha$.
- The above values are stored locally for future use.

Registration Phase. This phase aims to equip an entity requesting access to the system with the parameters necessary to operate in the following phases of the protocol. Specifically, in this phase the generic IoT sensor s_n registers to the system through the IoT Gateway g_k, and it receives the cryptography materials necessary to perform online authentication at run-time. The operations executed during the *Registration Phase* are depicted in Fig. 3.

- The IoT sensor s_n requests the registration to the system. To this aim, it delivers to the directly-connected gateway g_k its unique identifier s_n, the hashed username $H\left(U_n\right)$, and the hashed password $H\left(pwd_n\right)$. Note that PARFAIT does not require strictly the usage of username and password credentials, but also other forms of credentials can be used, such as biometric ones. However, for ease of exposition, the following description assumes the use of username and password credentials.
- The IoT Gateway g_k connects to the AtA, to register the IoT sensor s_n to the system. To this aim, it provides the unique identifier of the sensor s_n, the hashed username $H\left(U_n\right)$, and the hashed password $H\left(pwd_n\right)$, and further additional information, referred to as $info_n$. These pieces of information, added by g_k to the request, can include the registration type, the type of account, and further information useful to deduce the access rights of s_n.
- On receiving this information, the AtA stores the hashed username $H\left(U_n\right)$ and hashed password $H\left(pwd_n\right)$. Then, it forwards the ID of the sensor s_n and the additional information $info_n$ to the AzA.
- The AzA stores locally the ID s_n, and it maps the received information $info_n$ on a set of Z attributes $A_n = \left(a_{n,1}, \dots, a_{n,Z}\right)$, identifying the access rights of s_n. Then, the AzA delivers to the AtA the public parameters of the CP-ABE scheme. They include the bilinear groups $\mathcal{G}_0$ and $\mathcal{G}_1$, the generator γ, the prime order p of the group $\mathcal{G}_0$, the parameter $h = \gamma^\beta$, and the parameter $E = e(\gamma,\gamma)^\alpha$.

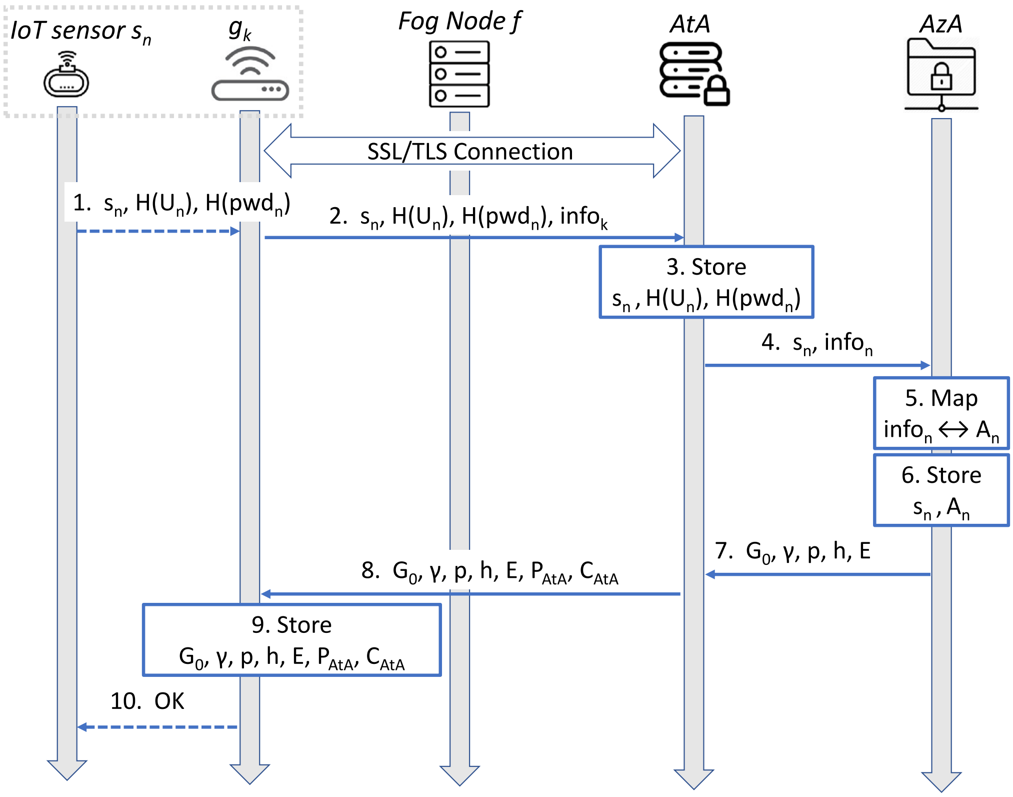

IoT sensor s_n
g_k
Fog Node f
AtA
AzA
SSL/TLS Connection
1. s_n, H(U_n), H(pwd_n)
2. s_n, H(U_n), H(pwd_n), $info_k$
3. Store s_n, H(U_n), H(pwd_n)
4. s_n, $info_n$
5. Map $info_n$ ↔ A_n
6. Store s_n, A_n
7. G_0, γ, p, h, E
8. G_0, γ, p, h, E, P_{AtA}, C_{AtA}
9. Store G_0, γ, p, h, E, P_{AtA}, C_{AtA}
10. OK

Fig. 3. Sequence diagram of the operations executed during the *Registration Phase* of PARFAIT.

- On the reception of the parameters from the AzA, the AtA forwards them to g_k, finalizing the registration phase. The message delivered from the AtA to the gateway g_k includes also the public key of the *AtA*, namely P_{AtA}, included in a X.509 certificate C_{AtA} signed by a Trusted Authority. Optionally, these parameters can be further forwarded to the IoT sensor s_n.

Note that, in line with the scenario and assumptions described in Section 4.1, the communication link between the IoT gateway and the Cloud services is secured via the well-known SSL/TLS protocol. Thus, all the messages exchanged in this phase are secured against eavesdropping and other active attacks (see Section 6).

Cloud Authentication Phase. This phase is executed online, when the IoT sensor(s) becomes operational. Overall, it involves the authentication of the IoT sensor(s) to the Cloud, and the delivery of cryptography materials from the Cloud to the IoT sensor(s), useful for accessing services in the fog-enabled ecosystem securely, for a limited time. The operations in this phase are summarized in Fig. 4 and described below.

- The IoT sensor s_n requests to the directly-connected gateway g_k to perform authentication with the Cloud, by delivering its unique id s_n, the hashed username $H\left(U_n\right)$, and the hashed password $H\left(pwd_n\right)$ over a secure connection.
- The gateway g_k performs the authentication with the AtA hosted in the Cloud, over a secure wireless connection, e.g., via the SSL/TLS protocol. To this aim, g_k delivers to the AtA the ID of the IoT sensor s_n, the hashed username $H\left(U_n\right)$, and the hashed password $H\left(pwd_n\right)$.
- The AtA verifies that the credentials provided by s_n through g_k match the ones locally stored and acquired during the *Registration Phase*. If there is a match, it forwards the identifier s_n of the requesting IoT sensor to the AzA. Otherwise, the request by g_k is discarded.
- The AzA gathers the information locally-stored about the attributes possessed by s_n, namely $A_n = \left(a_{n,1}, \dots, a_{n,Z}\right)$. Then, the AzA extracts J random bit-strings of size l, namely $\epsilon_n = \left[\epsilon_{n,1}, \dots, \epsilon_{n,J}\right]$. Each $\epsilon_{n,j}$ is an ephemeral identity, temporarily assigned by the AzA to the IoT sensor s_n. Note that, to speed up the lookup process in the following phases, the AzA could use dedicated algorithms and techniques, such as the ones described in [38]. However, being optional, such techniques are out of the scope of the manuscript.
The AzA also maps each ephemeral identity $\epsilon_{n,j}$ into an ephemeral attribute $\chi_{n,j}$, temporarily assigned to the IoT sensor s_n. The vector of ephemeral attributes is denoted as $\chi_n = \left[\chi_{n,1}, \dots, \chi_{n,J}\right]$.
- Then, for each ephemeral attribute $\chi_{n,j}$, the AzA creates an ephemeral secret key $SK_{n,j}$, by running the *Key Generation* procedure of the CP-ABE cryptographic scheme. Specifically, for each $\chi_{n,j}$, the AzA first concatenates $\chi_{n,j}$ with A_n, creating a vector $A_n^+ = \left(a_{n,1}, \dots, a_{n,Z}, \chi_{n,j}\right)$. Then, according to the CP-ABE scheme introduced in [30], it extracts a random $r \in Z_p$ and a number $Z+1$ of random values $r_z \in Z_p$, and it executes the operation in Eq. (1).

$$\begin{aligned} SK_{n,j} = \Big(& \delta = \gamma^{\frac{\alpha+r}{\beta}}, \\ & \forall z \in A_n^+ : \delta_z = \gamma^r \cdot H(z)^{r_z}, \delta'_z = \gamma^{r_z} \Big), \end{aligned} \tag{1}$$

where δ and δ_z are intermediate values.
The resulting vector is $SK_n = \left[SK_{n,1}, \dots, SK_{n,J}\right]$. The vector of ephemeral identities ϵ_n and the vector of secret keys $SK_n = \left[SK_{n,1}, \dots, SK_{n,J}\right]$ (one secret key for each ephemeral attribute/identity) is delivered from the AzA to the AtA. Note that the generation of the ephemeral attributes and ephemeral secret keys for the sensor s_n does not need to be executed at the time of the request from the sensor. Indeed, the AzA can pre-compute a given number of pairs SK_n, χ_n, and use them when it is necessary. This pre-computation is also feasible, given that

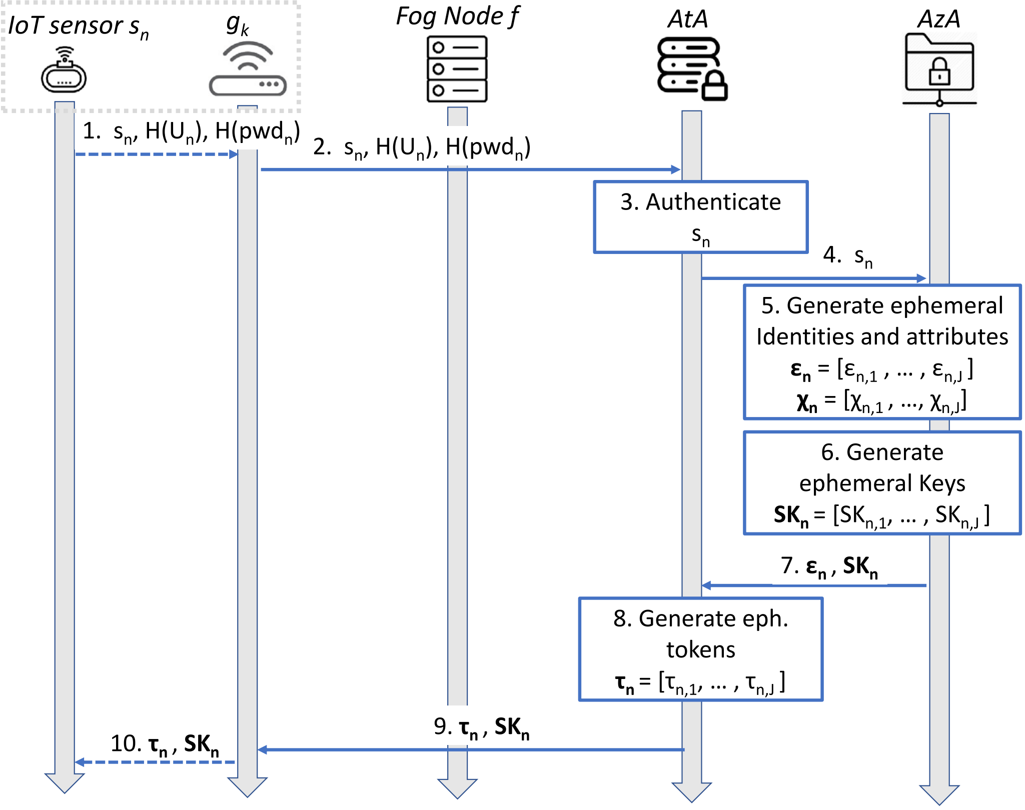

IoT sensor s_n
g_k
Fog Node f
AtA
AzA
1. s_n, H(U_n), H(pwd_n)
2. s_n, H(U_n), H(pwd_n)
3. Authenticate s_n
4. s_n
5. Generate ephemeral Identities and attributes
$\boldsymbol{\varepsilon_n}$ = [$\varepsilon_{n,1}$, … , $\varepsilon_{n,J}$]
$\boldsymbol{\chi_n}$ = [$\chi_{n,1}$, …, $\chi_{n,J}$]
6. Generate ephemeral Keys
$\mathbf{SK_n}$ = [$SK_{n,1}$, … , $SK_{n,J}$]
7. $\boldsymbol{\varepsilon_n}$, $\mathbf{SK_n}$
8. Generate eph. tokens
$\boldsymbol{\tau_n}$ = [$\tau_{n,1}$, … , $\tau_{n,J}$]
9. $\boldsymbol{\tau_n}$, $\mathbf{SK_n}$
10. $\boldsymbol{\tau_n}$, $\mathbf{SK_n}$

Fig. 4. Sequence diagram of the *Cloud Authentication Phase* of PARFAIT.

any entity in the CSP is equipped with large computational and storage capabilities.

- At the reception of the message from the AzA, for each of the ephemeral identity $\epsilon_{n,j}$ included in the message, the AtA creates an ephemeral token $\tau_{n,j}$. Each $\tau_{n,j}$ is a JWT, including a unique identifier uid, the iss field is the ID of the AtA, the sub field is the ephemeral identity $\epsilon_{n,j}$ assigned to the requesting IoT sensor s_n, the timestamp iat is the time when the token is created, namely $ts_{n,j}$, and the expiration date exp is $ts_{n,j} + T$, where T is a pre-defined validity time. The token also contains an additional claim, i.e., the claim eat, that contains the ephemeral attribute $a_{n,j}$. Finally, the token includes a signature $sgn_{n,j}$ generated by digitally signing the whole content of the token with the private key of the AtA, namely p_{AtA}, as in Eq. (2).

$$sign_{n,j} = E\left[H\left(\tau_{n,j} - \{sgn_{n,j}\}\right), p_{AtA}\right], \tag{2}$$

where $E[a, p]$ refers to a generic asymmetric encryption operation over the plain-text a, using the private key p, and the notation $\tau_{n,j} - \{sgn_{n,j}\}$ refers to the whole token, excluding the signature field $sgn_{n,j}$.

- Then, the AtA delivers to the gateway g_k the vector of ephemeral tokens $\tau_n = \left[\tau_{n,1}, \ldots, \tau_{n,J}\right]$, and the vector of ephemeral CP-ABE keys $SK_{n,j}$.

Fog Authentication and Authorization Phase. This phase occurs when the IoT sensor(s) would like to access services provided by the fog node(s). In this scenario, the IoT sensor shall first authenticate to the fog Node, and the process shall not involve Cloud-based services. In addition, without involving Cloud-based services, the IoT sensor should demonstrate to the fog Node to be authorized to access the requested service, i.e., to have a set of attributes satisfying the connected access policy. Finally, being the fog Node not fully trusted, the set of attributes and the identity of the requesting IoT sensor shall be hidden to the fog Node. The operations executed during this phase are summarized in Fig. 5 and explained below.

- Let us assume that the IoT sensor s_n would like to access the resource l_n hosted by the fog node f_q. To this aim, it forwards its unique ID s_n and the identifier of the resource l_n to the IoT gateway g_k, over a secure connection.
- The IoT gateway g_k extracts one of the valid and previously-unused ephemeral identities $\epsilon_{n,j}$ assigned to s_n, and the related ephemeral token $\tau_{n,j}$, and delivers the token to the fog node f_q, together with the identifier of the requested resource l_n.
- The fog node f_q first verifies to possess the resource l_n. If this is true, it evaluates if the token $\tau_{n,j}$ is valid. To this aim, it checks first the validity of the signature $sgn_{n,j}$, using the locally-stored copy of the public key of the AtA, namely P_{Ata}, as in Eq. (3).

$$D\left[sgn_{n,j}, P_{AtA}\right] == H\left(\tau_{n,j} - \{sgn_{n,j}\}\right), \tag{3}$$

where $D[a, P]$ refers to a generic asymmetric decryption operation over the cipher-text a, using the public key P, while $H(\cdot)$ refers to a generic hashing function. If the check in Eq. (3) is verified, the fog node checks if the expiration time exp is less than the actual time. If this check is also successful, the fog node blindly authenticates the requesting device, i.e., it authenticates the requesting device with the ephemeral identity $\epsilon_{n,j}$, based on the provisioning of the valid token $\tau_{n,j}$, demonstrating the previous Cloud-based authentication.
- The next step for the fog node is to check if the requesting device has a suitable set of access rights to access the requested resource l_n. Let us assume that the resource l_n has an access policy ρ_l, defined over the universe of attributes $\mathcal{A}$. First, using the ephemeral attribute $\chi_{n,j}$ contained in the field *eat* of the token $\tau_{n,j}$, the fog node creates a new ephemeral access policy $\rho'_{l,n,j} = \rho_l \wedge \xi_{n,j}$. Then, the fog node f_q extracts a random value $c_{l,n,j}$ and, according to the CP-ABE encryption scheme, it creates a *challenge* $\sigma_{l,n,j}$, by encrypting $c_{l,n,j}$ using the ephemeral access policy $\rho'_{l,n,j}$, as

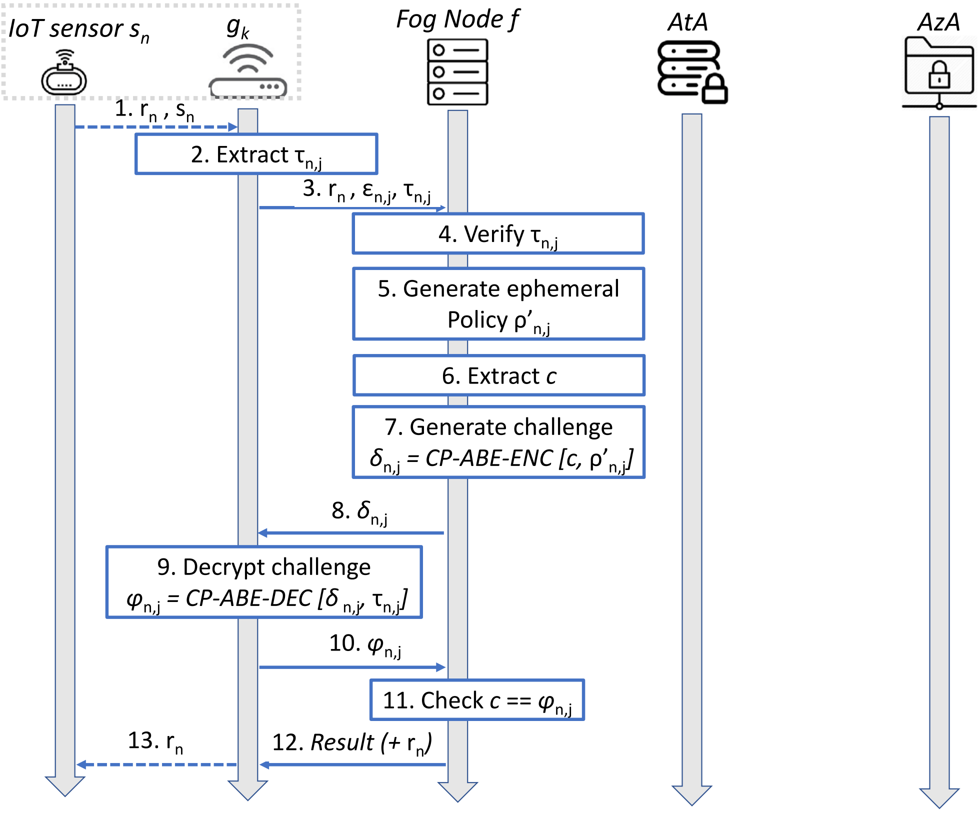

IoT sensor s_n
g_k
Fog Node f
AtA
AzA
1. r_n, s_n
2. Extract $\tau_{n,j}$
3. r_n, $\varepsilon_{n,j}$, $\tau_{n,j}$
4. Verify $\tau_{n,j}$
5. Generate ephemeral Policy $\rho'_{n,j}$
6. Extract c
7. Generate challenge $\delta_{n,j}$ = CP-ABE-ENC [c, $\rho'_{n,j}$]
8. $\delta_{n,j}$
9. Decrypt challenge $\varphi_{n,j}$ = CP-ABE-DEC [$\delta_{n,j}$, $\tau_{n,j}$]
10. $\varphi_{n,j}$
11. Check c == $\varphi_{n,j}$
12. Result (+ r_n)
13. r_n

Fig. 5. Sequence diagram of the *Fog Authentication and Authorization Phase* of PARFAIT.

specified in Eq. (4).

$$\sigma_{l,n,j} = CP-ABE-ENC\left(c_{l,n,j}, \rho'_{l,n,j}\right) = \left(\rho'_{n,j}, \tilde{c} = c_{l,n,j} \cdot e(\gamma,\gamma)^{\alpha c_{l,n,j}}, \bar{c} = h^{c_{l,n,j}},\right. \forall z \in A_n^+ : C_z = \gamma^{q_z(0)}, C'_z = H(z)^{q_z(0)}\Big), \tag{4}$$

where $e(\cdot)$ refers to the bilinear pairing operation ($\mathcal{G}_0 \times \mathcal{G}_0 \to \mathcal{G}_1$), while the notation $q_R(x)$ refers to the evaluation of the polynomial q_x of degree R in the point x. The resulting challenge $\sigma_{l,n,j}$ is then delivered to the gateway g_k.

- The IoT gateway g_k executes the challenge decryption on behalf of the IoT sensor s_n. Therefore, on the reception of the challenge $\sigma_{l,n,j}$, it uses the secret key $SK_{n,j}$ associated to the ephemeral token $\tau_{n,j}$ to decrypt the challenge, as in Eq. (5).

$$\varphi_{l,n,j} = CP-ABE-DEC\left(\sigma_{n,j}, \tau_{n,j}\right) = \frac{\tilde{c}}{\frac{\left(\bar{c}, \gamma^{\frac{\alpha+r}{\beta}}\right)}{e(\gamma,\gamma)^{rs}}} = \frac{c \cdot e(\gamma,\gamma)^{\alpha \cdot s}}{e\left(h^s, \gamma^{\frac{\alpha+r}{\beta}}\right)} \cdot e(\gamma,\gamma)^{r \cdot s} = \frac{c \cdot e(\gamma,\gamma)^{(\alpha+r)\cdot s}}{e(\gamma,\gamma)^{(\alpha+r)\cdot s}} = c_{l,n,j}. \tag{5}$$

Note that, in line with the main logic of the CP-ABE crypto-system, the gateway g_k can compute correctly the value $e(\gamma,\gamma)^{rs}$ only if the set of attributes contained in the token $\tau_{n,j}$ satisfy the policy $\rho'_{l,n,j}$. Otherwise, the gateway will obtain a value $e(\gamma,\gamma)^{r's}$, that is inconsistent (more details on the mathematical foundations of the CP-ABE crypto-system can be found in [30]). The resulting decrypted challenge $\varphi_{l,n,j}$, namely the *Response*, is then delivered to the fog node f_q.

- The fog node checks if the received response $\varphi_{l,n,j}$ matches the locally-stored random value $c_{l,n,j}$. If the two values match, it means that the requesting device possesses a set of attributes satisfying the ephemeral access policy $\rho'_{l,n,j}$, and that are also *fresh*, based on the expiration time of the token. Therefore, it grants access to the requested resource l_n, by providing its value in the response to the gateway g_k.
- The gateway g_k forwards the resource value to s_n.

Note that the fog node delivers the resource l_n to the gateway as a *plaintext* value, without any additional encryption operation. On the one hand, recall that the connection between the gateway and the fog node is secured via SSL/TLS. Therefore, any passive adversary trying to access the resource value by eavesdropping on the wireless communication channel would be unsuccessful. On the other hand, compared to the traditional use of the CP-ABE crypto-system, PARFAIT uses CP-ABE encryption and decryption operations only once in the time T, and not every time the resource is accessed. Then, the fog node sets up a secure connection with the gateway used by the sensor, and it uses that connection to continuously deliver the resource/service for the duration T included in the exp field of the token. In this way, the gateway has to perform a limited number of cryptography operations, and the IoT sensor can access the resource value immediately, with no further delay.

Considerations on Handover. Handover operations occur when the IoT sensor(s) move to a different location in a short time and need to continue using the services provided by the fog node. In this scenario, the newly connected fog node should re-authenticate the IoT sensor(s) and restore the online connection to the requested services as soon as possible, while also verifying that the IoT sensor(s) have the rights to execute the specific requested actions. Note that the operations executed in this scenario are exactly the same described before, for the *Fog Authentication and Authorization Phase*. Therefore, if the IoT sensor has one or more valid tokens, it provides a token to the new fog node, to obtain the required service. At the same time, the fog node only evaluates the information contained within the supplied token, and it does not need to interact with the Cloud. Also, note that the handover operation is fully privacy-preserving. Indeed, the fog node does not know which other fog nodes previously served the IoT sensor, and neither it can obtain such information by colluding with other fog nodes, given that the tokens supplied by the IoT sensors are different and unlinkable. More details on these properties will be provided in Section 6.

Considerations on Asynchronous Token Revocation. PARFAIT also easily supports asynchronous token revocation. Specifically, asynchronous token revocation can occur when the CSP detects that a token assigned to an IoT sensor has been compromised, e.g., it has been stolen or intentionally leaked to an unauthorized party. The CSP can use at least two possible strategies for token misuse identification.

The first detection mechanism is triggered by the owner of an IoT device, and it relies on *compromise reports*. Specifically, when the owner of an IoT device discovers the compromise of one or more of its devices (e.g., through the application of dedicated anti-virus software locally), he/she can report this event to the Cloud Service Provider (CSP), providing the details of the compromised device(s).

The second detection method is triggered by the CSP, and grounds on identifying anomalies in the usage patterns of the tokens. Indeed, we recall that, in line with the traditional fog computing architecture, the fog nodes periodically provide statistical reports to the CSP, including data regarding the accessed resources and the users interacting with them. It is reasonable to assume that the fog nodes also report to the CSP the hashes of the tokens used to access locally-hosted resources, together with usage timestamps. Thus, if one or more tokens have been used multiply in a short time at totally different locations (a clear indication of token compromise), the CSP can identify such event and proceed with token(s) revocation. Note that this is only one of the many possible *anomalous patterns* that the CSP can detect, and it can use other indicators of compromise.

When the CSP detects such an event, it inserts the uid field of the compromised token into a Token Revocation List (TRL), that is delivered in broadcast to all the fog nodes. At the reception of a token from a requesting IoT sensor, the fog node can check if the uid of the provided token is in the TRL. If there is a match, the fog node stops the processing of the token and rejects the request. Note that the TRL only contains the list of asynchronously revoked tokens, and it updates the list frequently, e.g., when the token expires. In this way, the CSP can take the size of the TRL short, resulting in limited storage, communication overhead, and processing time.

6. Security analysis

This section discusses the most important security properties of PARFAIT (Section 6.1) and provides a formal verification via Proverif (Section 6.2).

6.1. Security considerations

PARFAIT guarantees the following security properties.

- *IoT sensor(s) Authentication to the Cloud.* PARFAIT allows each IoT sensor s_n to carry out authentication to the Cloud. The authentication to the CSP is performed in the *Cloud Authentication Phase*, using the credentials of the IoT sensor obtained during the previous *Registration Phase*. Recall that: (i) the *Registration Phase* is executed offline, before the deployment of the IoT network, and (ii) the authentication credentials are delivered to the CSP by the IoT gateway over a wireless connection secured via the well-known TLS protocol. The robustness of TLS against eavesdropping and impersonation attacks has been demonstrated extensively in the literature [39,40], and these security proofs can be adopted also to demonstrate the security of the authentication process between the IoT sensor(s) and the Cloud.
- *IoT sensor(s) Anonymous Authentication to fog node(s).* As a result of a successful authentication with the CSP, the IoT sensor s_n receives multiple ephemeral tokens $\tau_{n,j}$, with $j \in [1, J]$, each connected to an ephemeral identity $\epsilon_{n,j}$. When requesting service from a fog node, the IoT sensor s_n extracts a random ephemeral identity, and uses the corresponding ephemeral token to authenticate to the fog node. Note that s_n limits the extraction process only to identities (tokens) that have not been used previously for other sessions with any fog node, in a way to fully preserve anonymity and privacy. On the one hand, the authentication of the IoT sensor is accomplished by verifying the authenticity and integrity of the token $\tau_{n,j}$. If the token has been signed by the AtA and the integrity of the signature is verified, the fog node *trusts* the AtA, and authenticates the IoT sensor. On the other hand, each ephemeral token $\tau_{n,j}$ provides authentication via an ephemeral identity $\epsilon_{n,j}$, contained in the token. The security of the JWT is based on the security of the cryptography mechanisms used to generate the signature. Usually, public-key encryption techniques are used, such as Rivest Shamir Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA), whose security has been proved extensively in the literature [41,42]. However, only the AzA in the Cloud can link an ephemeral token/identity to the device s_n, but this cannot be done by the fog node. In this sense, the authentication of the IoT sensor to the fog node is *anonymous*, as the fog node does not know the real identity of the requesting IoT sensor. Finally, note that the execution of the challenge–response protocol requires the user to prove the possession of the (cryptographic key corresponding to the) attributes in its possession, protecting against the stealing of such information by a malicious fog node.
- *Protection against Passive Eavesdropping and Impersonation.* The wireless connections from the IoT gateway to the fog Node and the Cloud are assumed to be secured via the TLS protocol. As mentioned before, the robustness of TLS against eavesdropping and impersonation attacks has been demonstrated extensively in the literature [39,40], and these security proofs can be adopted also to demonstrate the robustness of PARFAIT against passive eavesdropping and impersonation attacks.
- *Protection against fog node(s) Collusion and Tampering.* From the security perspective, the main weakness derived by the deployment of fog nodes is the enhanced vulnerability compared to Cloud services. Indeed, fog nodes are deployed closer to the user, often in unattended or partially-attended locations, and the chances that they are compromised is higher than Cloud services. Moreover, a malicious adversary could also deploy rogue fog nodes, and use them to infer private information.

  To protect against multiple colluding fog nodes, PARFAIT provides identity privacy (anonymity), attributes privacy, and location privacy to the IoT sensors. Indeed, as previously described, each service request issued to a fog node by an IoT sensor is generated via an ephemeral token, linked to an ephemeral identity, containing an ephemeral attribute. The usage of the ephemeral identity guarantees anonymity to the IoT sensor. Therefore, the fog node does not know which specific IoT sensor performs the requests, and cannot link multiple requests to the same IoT sensor. The impossibility to link multiple sessions also provide location privacy to the IoT sensors towards colluding fog nodes. Indeed, even assuming multiple fog nodes collude to track the IoT sensors or the user possessing them, it is not possible to associate one or more requests to a given IoT sensor, and thus, it is not possible to follow the path of the IoT sensor. Finally, the challenge–response scheme realized in the *Fog Authentication and Authorization Phase* protects the privacy of the attributes possessed by the IoT sensor. Assuming the response from the IoT sensor is correct, the fog node only knows that the set of attributes possessed by the IoT sensor satisfy the access policy associated to the requested resource, but it does not know the full set of attributes in possession of the IoT sensor. This property guarantees full attributes privacy to the IoT sensor.

- *Revocation of Compromised Identities/Tokens.* As a side security service, PARFAIT enables easy revocation of compromised identities and tokens, via the periodic download of the TRL by the fog nodes. When the AzA becomes aware of the leakage of an ephemeral identity (either via owner reports or via anomalies in tokens usage), it inserts the unique identifier of the corresponding ephemeral token in the TRL, that is periodically downloaded by the fog nodes. At the reception of a request from an IoT sensor, the fog nodes simply check if the identifier of the received token is in the latest TRL. If a match is found, the request is discarded. Similarly, if all the ephemeral identities/tokens associated to an IoT sensor are leaked, all of them are inserted in the TRL by the AzA, so that to stop the IoT sensor from accessing any resource.
  Note that malicious fog node could try to report false anomalies to the CSP, with the aim of revoking users' tokens. Such a behavior is not in our adversary model, as it would require the fog node to be active, while our assumption is the *honest-but-curious* model. In addition, such malicious reporting would only cause a DoS for the user, but it could not be used to gather users' private information. Given that PARFAIT aims primarily at protecting users' privacy when interacting with potentially-compromised fog nodes, protecting against DoS is out of scope.
- *Revocation of Attributes.* Flexible attribute revocation in CP-ABE systems is a well-known challenge in the literature, and this is also true for PARFAIT [43,44]. In PARFAIT, the easiest way to revoke an attribute is to revoke all the ephemeral tokens of all the devices possessing that attribute, so that they have to carry out a new *Cloud Authentication* phase and receive the new set of attributes. On the one hand, flexible attribute revocation in CP-ABE systems is an active research topic, and this is a problem affecting all the systems based on this cryptography technique. On the other hand, any solution addressing asynchronous attributes revocation can be integrated transparently in PARFAIT, as it can be adapted to integrate any implementation of the CP-ABE cryptography technique.
- *Protection against Collusion of IoT sensors.* Although not being the core objective, PARFAIT also allows detecting and rejecting collusion among IoT sensors, thanks to the usage of the CP-ABE cryptography scheme and the ephemeral tokens. Indeed, each token $\tau_{n,j}$ is coupled with a unique symmetric key $SK_{n,j}$, uniquely associated to the token, the ephemeral identity of the IoT sensors for which the token has been generated, and all the attributes associated to the actual identity of the IoT sensor. In addition, the token also contains a unique identifier uid and the ephemeral identity $\epsilon_{n,j}$, stored in the sub field. Given that the key $SK_{n,j}$ is associated to all attributes possessed by the IoT sensors, it is impossible for an IoT sensor to decouple a single attribute from a token, due to the security guarantees offered by CP-ABE. Indeed, the entity submitting the token can decrypt the challenge sent by the Fog node in the *Fog Authentication and Authorization Phase* only if it has the ephemeral key associated with the token and all the attributes. Thus, even if the token has been handed out to another colluding entity, without the key it is not possible to use it.
  However, assuming also the key of the overall token has been handed out together with the token, the fog node alone could not be able to detect the collusion attack. This occurs because PARFAIT protects IoT sensors and their users from potentially malicious fog nodes, and the vice-versa is not a design requirement. However, PARFAIT can still allow the detection of collusion among IoT sensors by allowing the fog node to check with the AzA if the entity submitting the token is the entity responsible for the token itself. Such a scheme would require the IoT sensor to authenticate with the AzA, unveiling its actual identity and allowing for the detection of the attack.

6.2. Security analysis via Proverif

To further show that PARFAIT does not alter the main security properties of the involved building blocks, we verified its main security properties also using the automated verification tool ProVerif. ProVerif [45] is a software tool developed by researchers at Inria and largely adopted by the scientific community to formally verify the security of cryptographic protocols, especially when already-proven building blocks are mixed to generate new schemes [35,46,47].

To verify the security of a protocol, Proverif assumes that the adopted cryptographic primitives are robust, and that the adversary can run the same cryptography algorithms of the legitimate entities, knowing some of the cryptography materials, as specified by the user during the programming phase. Once the protocol has been defined, Proverif verifies the security of the scheme against the powerful Dolev-Yao attacker model, so assuming that the adversary can read and modify messages on the fly, as well as injecting its own messages [48]. When an attack is found, Proverif also provides a list of the steps performed by the attacker to break the specific security properties.

PARFAIT has been implemented in ProVerif, and four main events have been identified.

- *begin_AtA*, indicating that the AtA is requested to start an instance of the *Cloud Authentication Phase* of PARFAIT by a requesting entity;
- *begin_Fog*: indicating that the fog node is requested to start an instance of the *Fog Authentication and Authorization Phase* of PARFAIT by a requesting entity;
- *end_GW_Cloud*: indicating that the IoT gateway completes successfully an instance of the *Cloud Authentication Phase* of PARFAIT with the requesting entity;
- *end_GW_Fog*: indicating that the Fog Node completes successfully an instance of the *Fog Authentication and Authorization Phase* of PARFAIT with the requesting entity;

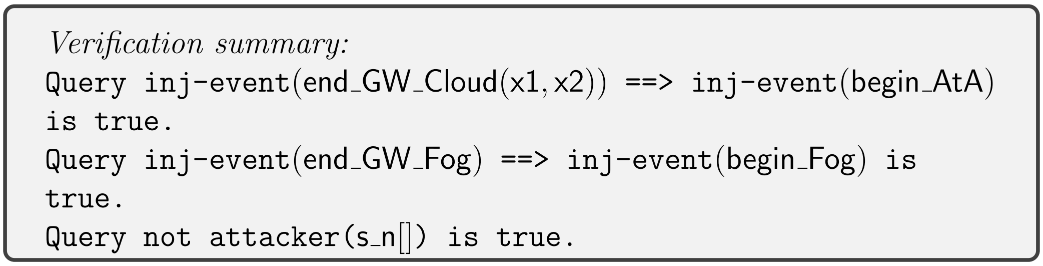
To correctly interpret the output of Proverif, we also recall that the tool provides the following output.

- *not attacker(elem[])*, when the adversary cannot capture the value of *elem*;
- *attacker(elem[])*, when the adversary can capture the value of *elem*;
- *inj-event(last_event ()) ==> inj-event(previous_event ()) is true*, when the function *last_event* is executed only when the function *previous_event* was really executed;
- *inj-event(last_event ()) ==> inj-event(previous_event ()) is false*, when it is not always true that when the function *last_event* is executed, the function *previous_event* was executed before.

To verify the security properties of PARFAIT, we run two different tests, whose output has been depicted in Figs. 6 and 7. Note that, to allow researchers to verify our claims, we also released the code of the implementation of PARFAIT in Proverif as open-source [49].

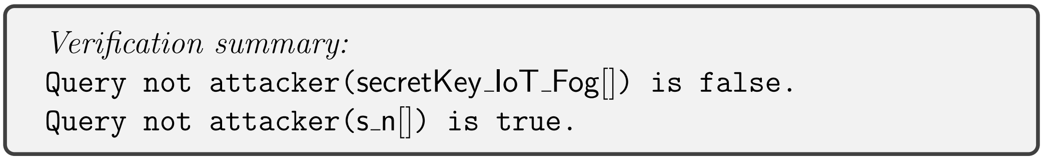
In the first test (1), we verified the first two properties listed in Section 6.1, i.e., the authentication of the IoT sensor to the Cloud and anonymous authentication of the IoT sensor to the Fog node. As you can see from the reported excerpt, every time that the event *end_GW_Cloud* is executed with the provided credentials, also the event *begin_AtA* was executed, meaning that the IoT gateway was the one starting the protocol. Similarly, every time that the event *end_GW_Fog* is executed, also the event *begin_Fog* was executed. Overall, these two claims prove the authentication to the two afore-mentioned security properties. In addition, with test (1) we also verified that the identifier s_n of the sensor node is never disclosed to the adversary, despite being delivered over the wireless channel. This proves the anonymity of the protocol to external eavesdroppers.

In the second test (2), we configured the fog node in a way to be compromised, by declaring as *public* the key used to secure the



```
Verification summary:
Query inj-event(end_GW_Cloud(x1, x2)) ==> inj-event(begin_AtA)
is true.
Query inj-event(end_GW_Fog) ==> inj-event(begin_Fog) is
true.
Query not attacker(s_n[]) is true.
```

Fig. 6. Excerpt of the output of Proverif with the protocol setup of test (1).



```
Verification summary:
Query not attacker(secretKey_IoT_Fog[]) is false.
Query not attacker(s_n[]) is true.
```

Fig. 7. Excerpt of the output of Proverif with the protocol setup of test (2).

communication channel between the fog node and the IoT gateway. As reported in the excerpt in Fig. 7, still the identifier of the sensor node is not available to the adversary. Thus, even if the fog node is malicious (i.e., it leaks all the materials received from the IoT gateway), the identifier s_n of the sensor node is never disclosed. Thus, the tampering of the fog node according to the honest-but-curious model does not provide any advantage to the adversary.

We remark that, in line with the literature adopting Proverif for protocol verification, the source code of PARFAIT in the ProVerif tool has been released as open-source [49], in a way to allow interested readers to verify our claim and further use our code as a ready-to-use basis for their software protocol verification.

7. Performance evaluation

This section describes the evaluation campaign of PARFAIT. Section 7.1 describes the implementation of the proof-of-concept, Section 7.2 shows the performance of PARFAIT, while Section 7.3 compares PARFAIT with related works.

7.1. Implementation details

The layer-2, layer-3, and layer-4 of PARFAIT have been implemented in a proof-of-concept in Python (version 3), using freely-available and open-source libraries. For the management and execution of the cryptography operations required by CP-ABE, the proof-of-concept adopts the latest version of the library *charm-crypto* (v0.50) [50]. It is a framework allowing quick prototyping of many recent crypto-systems, designed to reduce development time and foster code reproducibility. For the JWTs, the proof-of-concept uses the popular library *python-jwt* [51], while the popular *openssl* python library has been adopted for the creation, management, and validation of private/public key pairs and public key certificates [52].

Concerning the overall system architecture, the IoT gateway runs on a Raspberry PI version 3 model B+ equipped with the *Raspian Buster OS*, while the fog node and the CSP run on a DELL XPS 9560 laptop machine equipped with the Linux Ubuntu 20.04 LTS operating system and featuring 32 GB of RAM. Note that the hardware features of the Raspberry PI are fully in line with modern IoT gateways. At the same time, the adopted laptop represents a worst-case assumption for the fog node and the CSP. Indeed, fog-compliant devices and servers in a typical CSP farm are very likely to be more powerful than the used laptop, further reducing the reported computation times. Also, note that the requests originated from layer-1 IoT sensors have been simulated at the layer-2, through requests originated from the IoT gateway.

Two super-singular elliptic curves have been adopted for the CP-ABE operations, i.e., the curves $SS512$ and $SS124$, characterized by a group size of 512 bits and 1024 bits, respectively. Finally, the signatures of the JWTs have been computed through the $SHA - 256$ hashing algorithm and the well-known RSA scheme, due to its wider adoption [53].

7.2. Performance assessment

This section provides reports the delays and bandwidth required by PARFAIT to securely access resources.

The measurements of the delays in the communication between the IoT Gateway and the fog node have been carried out by exactly measuring the delay of the communication process between the Raspberry PI and the laptop. Conversely, given that the processes of the fog node and the CSP were located on the same machine, the performance assessment campaign adopted a methodology similar to the one adopted in [54]. Specifically, the communication latencies of the link between the IoT Gateway and the CSP have been modeled by introducing additional latencies in the communication process. These delays have been obtained by measuring the real end-to-end delay of the communication between two different hosts connected to the Internet. We identified ten different hosts, whose details and deployment has been summarized in Table 2, and we delivered a train of *ICMP Echo Requests* from the local network to each of the hosts. The corresponding communication latency has been measured as the absolute value of the difference between the time when the *ICMP Echo Requests* has been sent and the time when the corresponding *ICMP Echo Reply* has been received. The average value of the delay samples, as well as their 95% confidence intervals, are shown in Fig. 8. Therefore, any time the IoT Gateway needs to communicate with an entity on the CSP (and vice-versa), we select a reference host, extract a sample from the corresponding data set, and add it to the overall delay.

The first investigation reported hereby focuses on the time required to access resources hosted on the fog node. Specifically, the test assumes that initially, the IoT sensor does not possess the token necessary to access the resources hosted on the fog node, and therefore, the IoT sensor has to interact with the CSP before accessing resources. To generalize the investigation, the tests assumed the CSP to be located in each of the hosts reported in Table 2, and we measured the time necessary to: (i) authenticate on the CSP, (ii) obtain the ephemeral tokens, and (iii) submit one of the tokens to the fog node to access the resource (Phase #3 and Phase #4). As a common reference, in each test, $J = 10$ tokens are delivered by the AtA to the IoT gateway. The test has been repeated 1000 times for each host, configuring PARFAIT

Table 2
Details of the hosts used for the modeling of the communication process with the CSP.

ID	IP address	Nation	Location
$S1$	39.32.0.1	Pakistan	Islamabad
$S2$	8.8.8.8	USA	Mountain View
$S3$	76.74.224.13	Canada	Vancouver
$S4$	61.69.229.154	Australia	Sydney
$S5$	193.70.52.72	France	Paris
$S6$	167.71.129.73	England	London
$S7$	80.116.252.221	Italy	Rome
$S8$	202.46.34.59	China	Shenzhen
$S9$	125.30.18.121	Japan	Tokyo
$S10$	139.59.140.10	Germany	Frankfurt

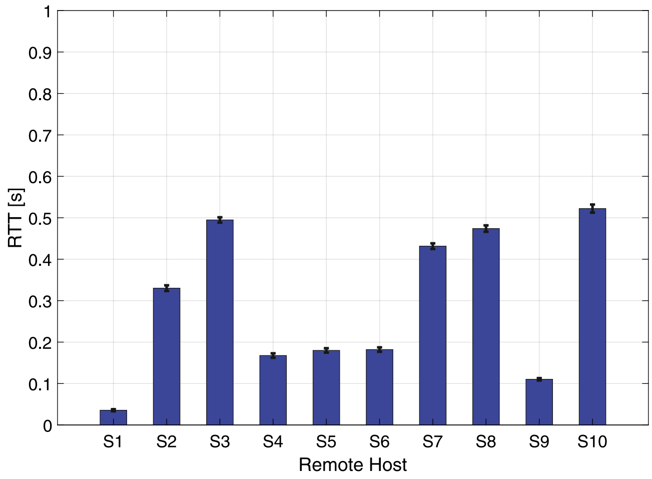


Fig. 8. Average RTT values measured for the remote hosts in Table 2.

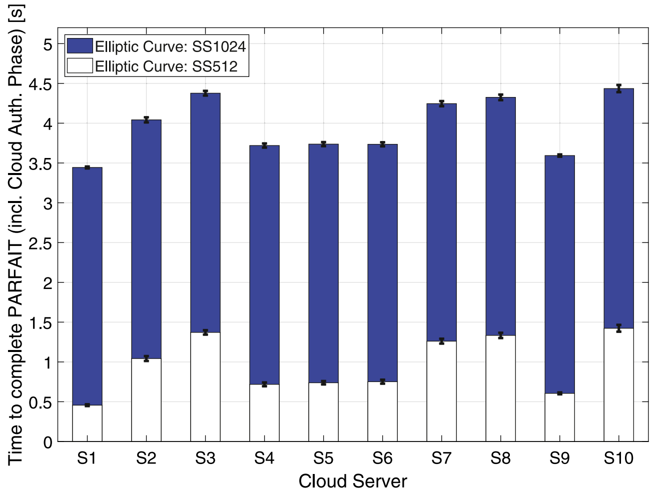


Fig. 9. Time to complete PARFAIT, including the Cloud Authentication Phase, assuming different CSP locations and two reference group sizes.

to use an elliptic group size of both 512 and 1024 bits, and the results are reported in Fig. 9. The height of each bar represents the mean value of all the measurements, reported together with the 95 % confidence interval of the measurements.

From the figure, it is possible to notice that the overall time to complete PARFAIT when no tokens are available mainly depends on two factors: (i) the group size, and (ii) the location of the CSP. The most influential feature is the group size; indeed, the larger the group, the higher the computation times on the fog node and the IoT gateway, and the larger the time to exchange the values necessary for the correct operation of the protocol. The shortest times are obtained with a group size of 512 bits and the server $S1$, with a mean value of 0.457 s. The time to complete the protocol with the same server $S1$ increases to 3.444 s when PARFAIT is configured with the group size of 1024 bits, but providing higher security. Conversely, the higher execution times were measured for the server $S10$, registering 1.423 s and 4.435 s when PARFAIT is configured with a curve of 512 bits and 1024 bits, respectively.

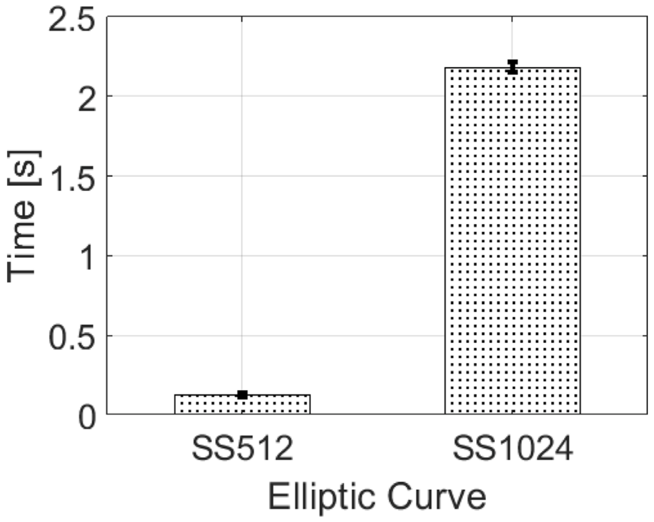


Fig. 10. Time to complete the *Fog Authentication and Authorization Phase* of PARFAIT, with two selected elliptic curves.

Recall that, once the tokens have been obtained, the IoT sensor does not need to access the CSP for every request, but it has to contact only the fog node hosting the resource. The time required for this process has been evaluated in Fig. 10, reporting the time required to complete only the *Fog Authentication and Authorization Phase* of PARFAIT, with the two selected elliptic curves. With $J = 10$ tokens, the *Fog Authentication and Authorization Phase* takes approx. 0.123 s and 2.182 s, when the elliptic curves $SS512$ and $SS1024$ are used, respectively. The performance jump in the usage of the curve $SS1024$ can be easily explained when looking at the architecture used for the IoT gateway, particularly slow with large group size. At the same time, the time to complete the *Cloud Authentication Phase* of PARFAIT also depends on the number of ephemeral tokens and ephemeral keys released by the CSP. The higher the number of tokens and keys, the larger the information to be transferred from the AtA to the IoT gateway, and the higher the time to complete the phase. Fig. 11 reports the time needed to complete the Phase #3, averaged on 1000 tests, varying the number of ephemeral tokens and keys released by the CSP. As a reference, these tests assumed that the IoT sensors interact with the server $S1$. The figure confirms that the time to complete the *Cloud Authentication Phase* increases linearly with the number of released tokens and keys. This result is due to two elements: (i) the time needed by the AtA to generate the tokens, and (ii) the time to transfer an increasing number of bytes. Note that the AzA does not need more time to generate new keys and ephemeral attributes, as these operations can be anticipated and executed before-hand, storing the ephemeral materials in a local database. However, also note that the tokens cannot be pre-computed, as the related generation timestamp (iat claim) and the expiration date (exp claim) have to be generated at the time of the request. Overall, assuming the use of a group size of 512 bits, PARFAIT takes 0.287 s when only one token (and the corresponding key) is released, while this time rises to 3.511 s when 100 tokens and keys are released. When a group of size 1024 bits is used, the above-mentioned latencies rise to 1.035 s and 4.262 s, respectively.

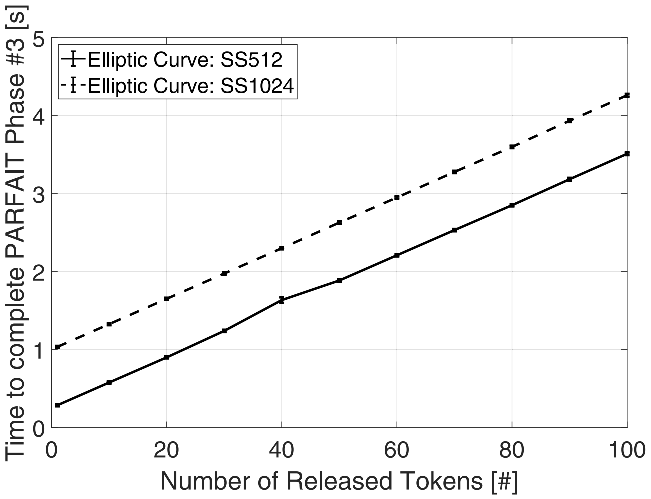


Fig. 11. Time to complete the Cloud Authentication phase of PARFAIT, with different number of ephemeral tokens and two reference group sizes.

From an architectural perspective, when a large number of computationally-constrained devices are included in the scenario, the computational and bandwidth overhead is on the IoT gateway. Indeed, for each request initiated by one of the IoT sensors, the gateway has to perform authentication and authorization procedures on the behalf of the computationally constrained devices. However, we notice that the operations in the *Cloud Authentication Phase* are executed only when the IoT sensor(s) become operational, and that the operations in the *Fog Authentication and Authorization Phase* are executed only once per request to access a given resource. Thus, even when a large number of IoT devices are connected to the IoT Gateway, the computational overhead for the gateway occurs only at the first resource requests, becoming regular soon after.

Finally, as for the processing and computational requirements, recent contributions such as [55] showed that pairing-based Attribute-Based Encryption algorithms could be also run efficiently on platforms even more constrained than the Raspberry-PI used in our manuscript (e.g., the ESP32 chip). For instance, assuming to work with a curve providing 80 bits level of security and considering the results in the cited publication with 5 attributes, the *Fog Authentication and Authorization Phase* of PARFAIT would take 6.039 seconds on the ESP32 device. Considering that such a delay is only present once in the connection (and not every time the service is requested), we believe it is a reasonable and acceptable delay, that does not affect too much the usability and applicability of PARFAIT even on more constrained devices. However, recalling that the gateway is usually more powerful than the leaf IoT sensors, it is always more efficient to let the IoT gateway execute CP-ABE operations. Such works contribute further to show that these techniques are feasible for (careful) integration in modern IoT devices.

7.3. Comparison with related work

This subsection compares PARFAIT against the proposals discussed in Section 2, both qualitatively (Section 7.3.1) and quantitatively (Section 7.3.2.

7.3.1. Qualitative comparison against competing solutions

Table 3 summarizes the related work in Section 2, providing a cross-comparison between PARFAIT and the discussed approaches, along the requirements discussed in Section 4.3. Note that most of the approaches considered fully-trusted fog nodes, used to outsource time-consuming and energy-demanding security operations. Despite improving the efficiency of the system in many use-cases, the above-mentioned strategy exposes the identity and attributes of the IoT sensors to the fog nodes. Thus, if the fog node is compromised, the adversary can fully impersonate the IoT sensors. At the same time, the adversary also knows private information, that can be sold to third-parties. At the same time, the only solution considering multiple colluding fog nodes, i.e., the proposal in [21], does not protect the privacy of the IoT sensors. Conversely, PARFAIT is the only solution able to address, at the same time, all the system and security requirements listed in Section 4.3. Indeed, PARFAIT protects both the anonymity and the privacy of the attributes of the IoT sensors, and it only uses the fog node to reduce latencies and delays, but not for accelerating or outsourcing security operations.

As possible weaknesses of PARFAIT, we mention that specific checks such as the ones for IoT sensors collusion and token revocation might imply sporadic interactions with the Cloud, thus increasing the access delay for the involved IoT devices. On the one hand, we recall that such operations might be executed only once per service access request, thus leading to limited performance impact. On the other hand, we notice that such issues are present also in the other works available in the literature, and are currently subject to further research activities for an efficient solution.

In addition, we notice that, in principle, users might still be tracked looking at the services they access across fog nodes. Investigating the unique relationship existing between users services request and their identity is also a future research direction.

7.3.2. Performance comparison

To provide further insights, this section also compares PARFAIT against some competing proposals, with reference to the time needed to access resources. We selected three competing solutions, characterized by a system architecture different from the one adopted by PARFAIT. The solution presented in [54] assumed that the resources are hosted on a Resource Server located on the Cloud, and that the user has to gather attributes from geographically-distributed attribute authorities before accessing the resource. Conversely, the solutions in [15,16] deployed resources on a local fog node, but they deliver always an encrypted version of the resource, using amendments of the CP-ABE algorithms. In turn, a dedicated fog node is used to decrypt the ciphertext and deliver the decrypted resource to the IoT sensor. Besides attempting at the privacy of the IoT sensors, such approaches also increase the aggregated time to access resources, especially with multiple resource requests. Indeed, each time the resource is required, the fog node delivers it encrypted, and the IoT sensor needs to deliver the ciphertext to the fog node for the decryption. Assuming the tokens and attributes have been already obtained, Fig. 12 shows the time required to access resources for PARFAIT and the above-mentioned approaches, considering an increasing number of resource requests. For the comparison with approaches involving Cloud interactions, the host $S1$ has been selected as the location of the CSP (most favorable case for the competing approaches, as it is the one reporting the shortest delays). Each point in the graph is the average of 1000 tests (95 % confidence intervals are also reported). We immediately notice that the approach in [16] (that is also adopted in [15]) leads to an excessive access delay, increasing quickly with the number of requests (0.192 s with a single request and 1.919 s with 10 requests, assuming the use of the curve with a group size of 512 bits). As previously highlighted, this situation occurs because each time a resource is needed, the IoT sensor needs to interact with a dedicated fog node for the decryption of the resource. Therefore, the gain obtained by running the CP-ABE decryption process on the fog node instead of the IoT gateway is compensated by the time required to access the fog node, degrading the performance. The solution in [54] reduces the number of interactions needed to access the resources, but still, it requires to access resources stored on the Cloud (0.402 s with a single request and 1.081 s with 10 requests, assuming the use of the curve with a group size of 512 bits). Conversely, PARFAIT takes full advantage of the fog computing architecture, deploying resources

Table 3
Qualitative comparison of PARFAIT against approaches for authentication and authorization in fog-enabled IoT ecosystems.

Ref.	Cloud-less fog authentication	Cloud-less fog authorization	IoT sensor(s) identity privacy	IoT sensor(s) attributes privacy	IoT sensor(s) location privacy	No collusion among IoT sensors	Protection against colluding fog nodes
[9]	✓	✗	✗	✗	✗	✗	✗
[10]	✓	✗	✗	✗	✗	✗	✗
[11]	✓	✗	✗	✗	✗	✓	✗
[12]	✗	✗	✗	✗	✗	✓	✗
[13]	✓	✗	✓	✗	✗	✗	✗
[14]	✓	✗	✓	✗	✗	✗	✗
[15]	✗	✗	✗	✗	✗	✓	✗
[16]	✗	✗	✓	✓	✗	✓	✗
[17]	✗	✓	✗	✗	✗	✓	✗
[18]	✗	✗	✗	✗	✗	✓	✗
[19]	✓	✓	✓	✓	✗	✓	✗
[20]	✓	✓	✓	✓	✗	✓	✗
[21]	✗	✓	✗	✗	✗	✓	✓
PARFAIT	✓	✓	✓	✓	✓	✓	✓

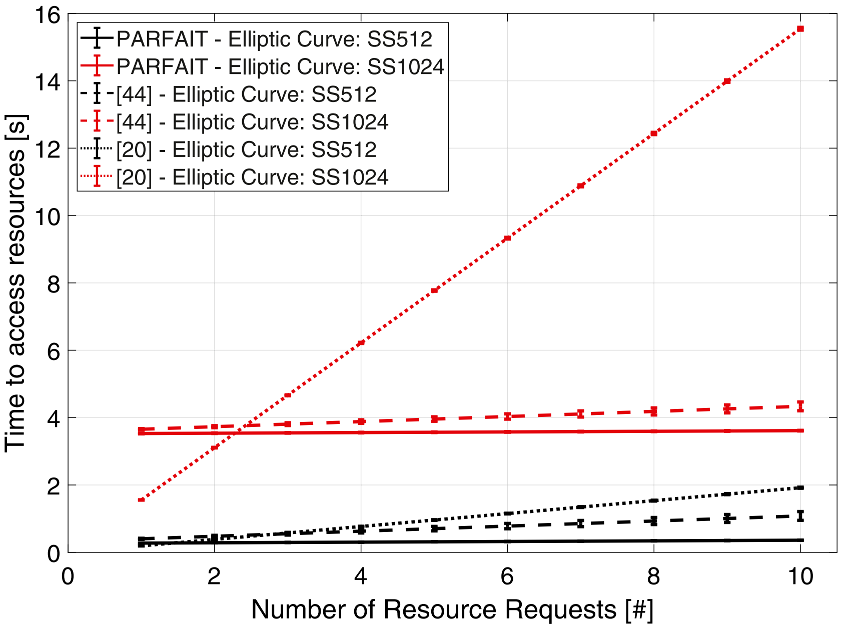


Fig. 12. Time required to access resources (tokens already obtained), with PARFAIT and the approaches in [16,54], with increasing requests.

closer to the requesting IoT sensors. At the same time, after the execution of the challenge–response protocol at the first resource request, PARFAIT uses the security services offered by lower-layer protocols (e.g., TLS), and delivers the requested resource without any additional encryption. Assuming the adoption of the curve with a group size of 512 bits, PARFAIT requires 0.274 s with a single request (very close to the average value obtained for the approach in [16]) and 0.359 s with 10 requests (81.2% less than the approach in [16] and 66.8% less than the approach in [54]). Only when the ephemeral token expires, the IoT sensor has to submit a new token and complete a new instance of the challenge–response protocol to access the requested resource. By selecting a suitable expiration time that trade-offs between usability and security, PARFAIT allows to fully leverage the advantages of the fog computing paradigm, while also protecting the privacy of the IoT sensors. Similar considerations also emerge when using the curve $SS1024$, and when IoT sensors generate multiple consecutive requests.

8. Conclusions

This paper presented PARFAIT, a privacy-preserving and low-delay security framework for fog-enabled IoT ecosystems. PARFAIT allows IoT devices to access resources stored on local fog nodes securely, while not requiring, at the same time, continuous interactions with authentication and authorization authorities hosted on the CSP. Moreover, different from related work, PARFAIT preserves the anonymity and the attributes privacy of the IoT devices from potentially untrusted fog nodes, resulting in a robust and reliable solution even in the presence of compromised fog nodes. Experiments ran on a dedicated proof-of-concept show that PARFAIT reduces the time needed to access resources, up to 81.2%, mostly when IoT devices perform multiple continuous resource requests—this scenario being prominent in modern IoT applications.

Security frameworks like PARFAIT, trading off performance with security and resilience, are becoming even more crucial to approach companies looking for enhanced network performances, raising awareness on the importance of protecting the privacy of sensitive data from ever-increasing cyber-attacks.

CRediT authorship contribution statement

Savio Sciancalepore: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by the INTERSECT project, The Netherlands, Grant No. NWA.1162.18.301, funded by Netherlands Organisation for Scientific Research (NWO). The findings reported herein are solely responsibility of the authors.

References

[1] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proc. First Edition Of The MCC Workshop On Mobile Cloud Computing, 2012, pp. 13–16.

[2] M. Mukherjee, L. Shu, D. Wang, Survey of fog computing: Fundamental, network applications, and research challenges, IEEE Commun. Surv. Tutor. 20 (3) (2018) 1826–1857.

[3] K. Tange, et al., A systematic survey of industrial internet of things security: Requirements and fog computing opportunities, IEEE Commun. Surveys Tuts. 22 (4) (2020) 2489–2520.

[4] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, fog others : A survey and analysis of security threats and challenges, Future Gener. Comput. Syst. 78 (2018) 680–698.

[5] P. Tedeschi, S. Sciancalepore, Edge and fog computing in critical infrastructures: Analysis, security threats, and research challenges, in: IEEE EuroS&PW, 2019, pp. 1–10.

[6] OpenFog Consortium, OpenFog Reference architecture for fog computing, 2017, https://tinyurl.com/y6au9pso. (Accessed 18 October 2021).

[7] A. Alwarafy, K. Al-Thelaya, M. Abdallah, et al., A survey on security and privacy issues in edge computing-assisted internet of things, IEEE Internet Things J. (2020).

[8] M. Caprolu, R. Di Pietro, F. Lombardi, S. Raponi, Edge computing perspectives: architectures, technologies, and open security issues, in: IEEE Int. Conf. Edge Comput., 2019, pp. 116–123.

[9] A. Ali, et al., Transparent 3rd-party authentication with application mobility for 5G mobile edge computing, in: IEEE EuCNC, 2020, pp. 219–224.

[10] F. Dewanta, M. Mambo, A mutual authentication scheme for secure fog computing service handover in vehicular network environment, IEEE Access 7 (2019) 103095–103114.

[11] X. Jia, D. He, N. Kumar, K.-K.R. Choo, A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing, IEEE Syst. J. 14 (1) (2019) 560–571.

[12] C. Wang, Y. Zhang, X. Chen, et al., SDN-Based handover authentication scheme for mobile edge computing in cyber-physical systems, IEEE Internet Things J. 6 (5) (2019) 8692–8701.

[13] M. Wazid, A.K. Das, N. Kumar, A.V. Vasilakos, Design of secure key management and user authentication scheme for fog computing services, Future Gener. Comput. Syst. 91 (2019) 475–492.

[14] P. Gope, LAAP: LIghtweight anonymous authentication protocol for D2D-aided fog computing paradigm, Comput. Secur. 86 (2019) 223–237.

[15] P. Zhang, Z. Chen, J. Liu, et al., An efficient access control scheme with outsourcing capability and attribute update for fog computing, Future Gener. Comput. Syst. 78 (2018) 753–762.

[16] K. Fan, H. Xu, L. Gao, H. Li, Y. Yang, Efficient and privacy preserving access control scheme for fog-enabled IoT, Future Gener. Comput. Syst. 99 (2019) 134–142.

[17] K. Fan, et al., A secure and verifiable outsourced access control scheme in fog-cloud computing, Sensors 17 (7) (2017) 1695.

[18] S. Tu, M. Waqas, F. Huang, G. Abbas, Z.H. Abbas, A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing, Comput. Netw. 195 (2021) 108196.

[19] S. Xu, J. Ning, J. Ma, X. Huang, H.H. Pang, R.H. Deng, Expressive bilateral access control for internet-of-things in cloud-fog computing, in: Proceedings Of The 26th ACM Symposium On Access Control Models And Technologies, in: SACMAT '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 143–154.

[20] S. Xu, J. Ning, Y. Li, Y. Zhang, G. Xu, X. Huang, R. Deng, Match in my way: Fine-grained bilateral access control for secure cloud-fog computing, IEEE Trans. Dependable Secur. Comput. (2020).

[21] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, X. Cheng, An attribute-based encryption scheme to secure fog communications, IEEE Access 5 (2017) 9131–9138.

[22] F. Alharbi, A. Alrawais, A.B. Rabiah, S. Richelson, N. Abu-Ghazaleh, CSProp: Ciphertext and signature propagation low-overhead public-key cryptosystem for IoT environments, in: 30th {USENIX} Security Symposium ({USENIX} Security 21), 2021, pp. 609–626.

[23] K. Rantos, G. Drosatos, K. Demertzis, C. Ilioudis, A. Papanikolaou, Blockchain-based consents management for personal data processing in the IoT ecosystem, in: ICETE, Vol. 2), 2018, pp. 738–743.

[24] A. Almohaimeed, S. Gampa, G. Singh, Privacy-preserving IoT devices, in: IEEE Long Island Systems, Applications And Technology Conference, LISAT, 2019, pp. 1–5.

[25] W. Bao, D. Yuan, Z. Yang, et al., Follow me fog: Toward seamless handover timing schemes in a fog computing environment, IEEE Commun. Mag. 55 (11) (2017) 72–78.

[26] T.N. Gia, A.M. Rahmani, T. Westerlund, P. Liljeberg, H. Tenhunen, Fog computing approach for mobility support in internet-of-things systems, IEEE Access 6 (2018) 36064–36082.

[27] W. Bao, et al., SFog: Seamless fog computing environment for mobile IoT applications, in: ACM Int. Conf. On Modeling, Analysis And Simulation Of Wirel. And Mob. Sys., 2018, pp. 127–136.

[28] M. Palattella, R. Soua, A. Khelil, et al., Fog computing as the key for seamless connectivity handover in future vehicular networks, in: Proc. ACM Symp. On Applied Computing, 2019, pp. 1996–2000.

[29] M. Jones, et al., JSON Web Token (JWT), RFC 7519 Tech. Rep, 2015.

[30] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium On Security And Privacy, SP '07, 2007, pp. 321–334.

[31] K. Emura, A. Miyaji, A. Nomura, et al., A ciphertext-policy attribute-based encryption scheme with constant ciphertext length, in: Int. Conf. On Informat. Security Practice And Experience, 2009, pp. 13–23.

[32] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: Int. Colloquium On Automata, Languages, And Programming, 2008, pp. 579–591.

[33] Z. Zhou, D. Huang, On efficient ciphertext-policy attribute based encryption and broadcast encryption, in: Proc. ACM Conf. On Computer And Communications Security, 2010, pp. 753–755.

[34] C. Bormann, M. Ersue, A. Keranen, Terminology for constrained-node networks, in: Internet Engineering Task Force, IETF, Fremont, CA, USA, 2014, pp. 2070–1721.

[35] P. Tedeschi, S. Sciancalepore, A. Eliyan, R. Di Pietro, Like: Lightweight certificateless key agreement for secure IoT communications, IEEE Internet Things J. 7 (1) (2019) 621–638.

[36] S. Sciancalepore, G. Piro, G. Boggia, et al., Public key authentication and key agreement in IoT devices with minimal airtime consumption, IEEE Embed. Syst. Lett. 9 (1) (2016) 1–4.

[37] Q. Zhu, R. Wang, Q. Chen, Y. Liu, W. Qin, Iot gateway: Bridging wireless sensor networks into internet of things, in: 2010 IEEE/IFIP International Conference On Embedded And Ubiquitous Computing, Ieee, 2010, pp. 347–352.

[38] M. Khodaei, et al., Scaling pseudonymous authentication for large mobile systems, in: Proc. Of ACM WiSec, 2019, pp. 174–184.

[39] H. Krawczyk, et al., On the security of the TLS protocol: A systematic analysis, in: Annual Cryptology Conf., 2013, pp. 429–448.

[40] T. Jager, F. Kohlar, S. Schäge, et al., On the security of TLS-DHE in the standard model, in: Annual Cryptology Conf., 2012, pp. 273–293.

[41] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.

[42] I. Blake, G. Seroussi, G. Seroussi, N. Smart, Elliptic Curves in Cryptography, Vol. 265, Cambridge University Press, 1999.

[43] S. Yu, C. Wang, K. Ren, W. Lou, Attribute based data sharing with attribute revocation, in: Proc. 5th ACM Symposium On Information, Computer And Communications Security, 2010, pp. 261–270.

[44] J. Li, W. Yao, J. Han, Y. Zhang, J. Shen, User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage, IEEE Syst. J. 12 (2) (2017) 1767–1777.

[45] B. Blanchet, Automatic verification of correspondences for security protocols, J. Comput. Secur. 17 (4) (2009) 363–434.

[46] C. Cremers, L. Hirschi, Improving automated symbolic analysis of ballot secrecy for E-voting protocols: A method based on sufficient conditions, in: 4th IEEE European Symposium On Security And Privacy, EuroS&P'19, 2019.

[47] T. Antignac, M. Mukelabai, G. Schneider, Specification, design, and verification of an accountability-aware surveillance protocol, in: Proceedings Of The Symposium On Applied Computing, in: SAC '17, 2017, pp. 1372–1378.

[48] I. Cervesato, The dolev-yao intruder is the most powerful attacker, in: 16th Annual Symposium On Logic In Computer Science—LICS, Vol. 1, 2001.

[49] S. Sciancalepore, Open-source code of the implementation of PARFAIT in the the ProVerif tool, 2021, https://github.com/ssciancalepore/parfait-proverif. (Accessed 05 October 2021).

[50] JHU Security and Crypto Lab, Charm: A framework for rapidly prototyping cryptosystems, 2021, https://github.com/JHUISI/charm. (Accessed 18 October 2021).

[51] Gehirn Inc., JSON Web token library for Python 3, 2021, https://pypi.org/project/jwt/. (Accessed 18 October 2021).

[52] OpenSSL, Python wrapper module around the OpenSSL library, 2021, https://pypi.org/project/pyOpenSSL/. (Accessed 18 October 2021).

[53] R. van Rijswijk-Deij, M. Jonker, A. Sperotto, On the adoption of the elliptic curve digital signature algorithm (ECDSA) in DNSSEC, in: 12th International Conference On Network And Service Management, CNSM, IEEE, 2016, pp. 258–262.

[54] S. Sciancalepore, G. Piro, D. Caldarola, et al., On the design of a decentralized and multiauthority access control scheme in federated and cloud-assisted cyber-physical systems, IEEE Internet Things J. 5 (6) (2018) 5190–5204.

[55] P. Perazzo, F. Righetti, M. La Manna, C. Vallati, Performance evaluation of attribute-based encryption on constrained IoT devices, Comput. Commun. 170 (2021) 151–163.

Savio Sciancalepore is currently Assistant Professor in IoT security at Eindhoven University of Technology (TU/e), Eindhoven, Netherlands. He received his master degree in Telecommunications Engineering in 2013 and the Ph.D. in 2017 in Electric and Information Engineering, both from the Politecnico di Bari, Italy. He received the prestigious award from the ERCIM Security, Trust, and Management (STM) Working Group for the best Ph.D. Thesis in Information and Network Security in 2018. From 2017 to 2020, he was Post Doc Researcher at HBKU-CSE-ICT, Doha, Qatar. His major research interests include network security issues in Internet of Things (IoT) systems and Cyber–Physical Systems, including UAV networks, avionics systems, and mobile networks.