Privacy-Preserving Fog Computing Paradigm

Nabil Abubaker, Leonard Dervishi and Erman Ayday
Department of Computer Engineering
Bilkent University
Ankara, Turkey
{nabil.abubaker, leonard.dervishi}@bilkent.edu.tr, erman@cs.bilkent.edu.tr

***Abstract*—As an extension of cloud computing, fog computing is proving itself more and more potentially useful nowadays. Fog computing is introduced to overcome the shortcomings of cloud computing paradigm in handling the massive amount of traffic caused by the enormous number of Internet of Things devices being increasingly connected to the Internet on daily basis. Despite its advantages, fog architecture introduces new security and privacy threats that need to be studied and solved as soon as possible. In this work, we explore two privacy issues posed by the fog computing architecture and we define privacy challenges according to them. The first challenge is related to the fog's design purposes of reducing the latency and improving the bandwidth, where the existing privacy-preserving methods violate these design purposed. The other challenge is related to the proximity of fog nodes to the end-users or IoT devices. We discuss the importance of addressing these challenges by putting them in the context of real-life scenarios. Finally, we propose a privacy-preserving fog computing paradigm that solves these challenges and we assess the security and efficiency of our solution.**

***Index Terms*—Privacy in Fog, IoT, Fog Computing**

I. Introduction

Cloud computing has made possible the sharing of storage and processing power between many devices in a centralized manner. It makes possible the storage and processing of data provided by end-users and smart devices in a fast way. After the introduction of Internet of Things (IoT), many extra devices or "Things" will be connected to the Internet amplifying the current number of connected devices to higher magnitudes. This increase will affect the current cloud-based computing architecture with the condensed traffic and will increase latency, limit bandwidth, and might seriously affect the Quality of Service (QoS). Furthermore, many IoT-based applications, such as industrial internet and health monitoring, require real-time responses in milliseconds and high reliability. Cloud computing, under the umbrella of IoT, lacks the necessary tools to deal with such applications [1]. For this reason, a new platform was introduced, which could handle such problems: fog computing. In this new platform fog nodes are introduced, which lie at the edges of the network. The location of the fog nodes is near the IoT devices and that makes it more adequate in terms of bandwidth usage and computation time [2]. Thus, in principal fog computing is relatively a new topic devised mainly to overcome the overhead and latency problems caused by the IoT. As a platform, it is based on cloud computing and for that reason it also inherits some unsolved security and privacy matters.

978-1-5386-0683-4/17/$31.00

Data privacy nowadays is a very hot topic. During the past two decades, an extensive amount of research has been done in order to limit the control of big companies and governmental authorities over users' data by devising and improving privacy-preserving algorithms and protocols. The privacy debate has been going for a long time, but recently, advancement of technology has made it easier for companies to collect more information about customers in order to increase their profit. Furthermore, for the same reason, governments' job of collecting information about citizens is made easier. The aim of privacy-preserving techniques is to increase the users' control over their own private data and prevent it from being disclosed to any party without their consent. There are several privacy and security challenges under the cloud computing paradigm and there exists a lot of proposed solutions for them in the literature. Most of these challenges are inherited to the fog computing architecture and some of their existing solutions can be applied to the fog architecture. However, in this work we will not be addressing these issues. Rather, we will concentrate on new privacy challenges that are introduced by the fog architecture and cannot be solved by the existing privacy-preserving techniques.

In this work, we address two privacy issues and challenges that are related to the nature of fog architecture and cannot be solved using existing privacy-preserving methods. The first privacy issue is related to the fog's design purpose of reducing the latency and improving the bandwidth. The second issue is related to the nature of fog architecture where the fog nodes reside in a location close to the end-user or IoT device. We discuss how this closeness can be used to disclose the location of the end-user, even if a location obfuscation method is applied. Then we propose a paradigm of fog computing that has two added generally-defined elements and we propose instances of those elements. We define a privacy-preserving protocol for each of those instances and we assess the efficiency and security of the proposed protocols.

The rest of this paper is organized as follows: Background information about IoT, fog computing, and why fog computing is designed for IoT are presented in Section II. In Section III, we present the fog architecture and some assumptions that we consider throughout this work. In Section IV, we present the privacy issues and challenges posed by the fog architecture and we give example scenarios where those challenges become

significant. We propose solutions to the privacy challenges defined in Section IV, and we conduct a discussion about their validity and applicability in Sections V and VI respectively. Related works and how they differ from our work are given in Section VII. In Section VIII the paper is concluded.

II. Background

In this section, we explain the background concepts of IoT, fog computing, and some related privacy issues between fog nodes and end-users in fog computing.

A. Internet of Things (IoT)

The Internet of Things (IoT) is the network of smart devices, houses, vehicles, and other similar products which are interconnected with sensors and actuators. There is also a network connection which makes possible the exchange of data between them and other nodes of the system. IoT makes possible the integration of devices within computerized system [3]. A misconception about "things" in IoT, is that it does not refer just to smartphones or computers but also to diverse set of devices such as sugar monitoring device, home sensors, and automated vehicles.

B. Fog Computing

As previously mentioned, fog computing is a new platform which has been spreading in a fast way because of the problems it promises to overcome such as the overhead and latency problems of the IoT working under cloud computing.

To have a better understanding of fog computing, we first describe the reasons that led to the need of this new paradigm. Afterwards, we give a general overview of fog computing, together with real life examples on its usage.

1) Derivation from cloud computing: Cloud computing has proven itself very useful since its beginning. As many other platforms, it also has its weak points and one of them is the large distance between the cloud and the IoT devices. The growing usage of IoT devices made it difficult to satisfy all demands of mobility support. Also, it introduces network overhead and high latency problems as the number of IoT devices is huge compared to the normal personal computers and smart phones currently connected to the Internet.

In order to resolve these kinds of problems, another platform which would bring storage and computation of data close to the edges of the network has been proposed. The new platform derived from cloud computing is called fog computing and the major change is the addition of fog nodes near IoT devices [4].

There are different paradigms derived from cloud computing such as Edge Computing, Mobile Cloud Computing (MCC), and Mobile Edge Computing (MEC). The surveys [5], [6] present a good taxonomy of those paradigms and explore their security and privacy threats.

2) Fog computing overview: As previously stated, fog computing makes possible services such as storage and computational power to be on the edge of the network (hence, closer to the end-user). Vaquero [7] defines the fog as a huge number of heterogeneous devices communicate with each other and cooperate to achieve storage and processing tasks, as services, in a decentralized manner. The devices in this definition (fog nodes) can be corporate-controlled computers distributed in different areas to provide hardware as a service (HaaS), or they can be part of personal computers of ordinary citizens, where they lease a part of their computers to host services and get incentives for it.

The way this platform works can be described as follows: there are a group of IoT devices near each fog node (see Fig. 1). An IoT device may be connected to one or multiple fog nodes. Based on the availability of fog nodes, IoT devices send their data to one or multiple nodes to be processed. The selection of the IoT device-fog node pairs is done mainly based on their distance from each other.

3) Fog computing in real-life: Fog computing is currently used in few areas and in this subsection only two of them are described: smart grid and smart traffic lights [4]. In smart grid there is an application which makes possible the switch between solar and wind energy, based on information given by appropriate sensors. Fog nodes collect data from the sensors and they process that data. If needed, they filter the data taken from the sensors and send it to cloud where further computations or analytics are done. As it can be observed in the smart grid, there is a close relation between cloud and fog nodes to decide about the switching of energy. Another case is smart traffic lights, where light sensors on the street can easily detect ambulance's flashing lights and change the state of the traffic light. Sensors can also detect pedestrian movements and send that information to the fog nodes which may decide to change the traffic light state if the amount of time for vehicles is over, or keep it as it is if it does not detect anything.

C. Current Privacy Issues in Fog Computing

As mentioned in [8], [9], there are known privacy issues in fog computing:

1) Data privacy: In fog computing architecture, data privacy is at risk because of the fog nodes positions (near to end-users) [10]. This makes it vulnerable to collect more sensitive information compared to the cloud architecture. Furthermore, in fog computing customer data is outsourced to the fog node. Therefore, fog nodes can collect the data from IoT devices and relate them with the real identities of the clients.

2) Usage patterns: This mainly refers to the frequency of data being sent to the fog nodes (the usage pattern of IoT devices). A good example for this one is smart grid, where the adversary can infer sensitive information by checking the usage of electricity or idle time of the doors.

3) Location privacy: The location of end-users is at risk because of the spatial correlation between fog nodes and IoT devices. Since the clients generally assign their tasks to the nearest fog node, it can be deduced that the client is closer to that node and away from others. Therefore, the location of the IoT devices should be hidden from the fog nodes considering the possible adversarial behaviour of them.

III. Architecture and Assumptions

In this section, we define the fog computing architecture as well as some assumptions we consider in this work.

A. Architecture Definition

The fog architecture used in our work is the same as defined in [3], which has 4 main layers: IoT devices layer, the fog nodes layer, the aggregate fog nodes layer, and the cloud layer. The IoT devices layer may consist of all kinds of IoT devices for all applications. Throughout this paper, the terms "end-user", "IoT device", and "fog client" all represent elements of this layer.

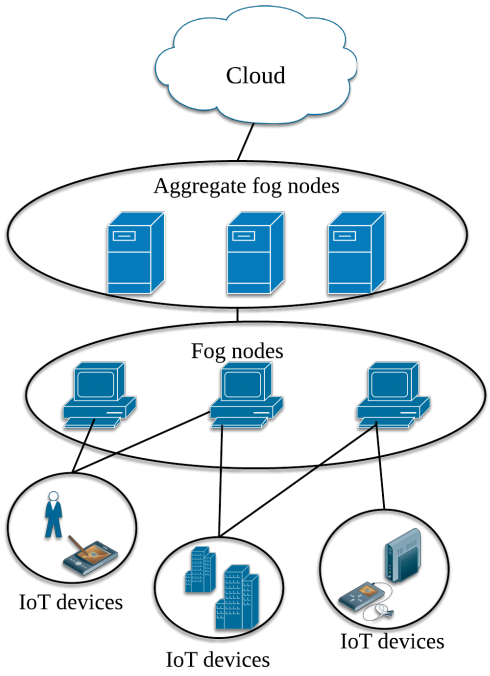

Fig. 1. Fog computing architecture.

The fog nodes layer consists of machines that are capable of performing small to medium computations. Example of such machines are personal computers and small workstations. This layer is used to handle the majority of computations required by the IoT devices.

The aggregate layer is used for handling aggregate data reported by the fog layer and support the fog layer with its higher computational powers. Furthermore, large scale computations that cannot be handled by the fog nodes and are not time-critical are usually offloaded to the aggregate layer.

The cloud layer is responsible for the ultimate and long-time storage, handling large scale computations as well as managing the nodes in the other layers. Fig. 1 shows the fog architecture used in our work.

B. Assumptions

We make the following assumptions throughout this work:

- The IoT devices are capable of handling encryption and decryption operations. Usually, the IoT devices are of low computational power, but it is convenient to assume that most of the devices used outdoors like smart cars systems or smart watches (or any kind of wearables) are capable of handling cryptographic operations that are required by majority of protocols such as public key encryption and symmetric encryption. Furthermore, for indoor devices that do not have the computational power to encrypt or decrypt data such as sensors, an "IoT node" that have a fair computational capability can be added for the house or facility to handle all types of traffic coming in or out of the house/ facility. This IoT node takes the responsibility of encapsulating the sensors data in an TCP/IP packet and handle all required cryptographic operations.
- The fog nodes and aggregate fog nodes are placed in secure locations, and the communication between them and with the cloud is secured via end-to-end encryption.
- Even though it is possible for an IoT device to use the services of several fog nodes, for simplicity we assume that the closest fog node to the IoT device is always responsible for handling its tasks.
- The mixes and trusted third party nodes (to be introduced in Section V) are controlled by a trusted authority other than the cloud authority, and they do not collude with the fog nodes or the cloud to disclose information about the users.

IV. Privacy Issues and Challenges Posed by the Fog Architecture

It is possible to apply privacy preserving techniques used in the literature on the fog architecture. However, there are some additional privacy issues as a consequence to its design. In this section, we explore two of them and we define corresponding privacy challenges that need to be addressed and solved.

A. Shortcomings of Existing Privacy-Preserving Methods

The definition of the fog architecture contradicts with the basic requirements of some privacy-preserving methods in the literature. For instance, conducting the computation required by an IoT device at the nearest fog node in order to minimize latency and conserve network bandwidth contradicts with the requirement of several privacy-preserving methods that require the interaction with one or more far-away third party to provide anonymity. As an example, if one wishes to achieve anonymity, using proxy server, VPN, or Tor (onion routing) are not applicable under the scope of fog computing because they introduce some latency as they require communication with distant parties.

The challenge related to this issue is **hiding the identity of the IoT device from the fog node**. This challenge is related to the first issue because hiding the identity of the IoT node needs the communication with a distant third party, which contradicts with one or more of the design purposes of the fog architecture (conduct computations at the nearest node, reduce latency and bandwidth). This challenge cannot be solved using the existing privacy-preserving methods under the umbrella of fog computing. Hiding the true identity of

the client requires the involvement of a third party in any existing privacy-preserving protocol. This means that more than 50% of the communication has to be through a distant third party, hence, the design purpose of reducing the latency by conducting the computations at the nearest fog node is violated.

There are several scenarios reflecting the need of a solution for this challenge:

- Facilities that use fog services to control their machines might want to hide their usage patterns by hiding their identities.
- A smart lock needs identity obfuscation because no one wants any outsider to know if they are opening/ closing their doors. Note that in this case, knowing the identity of the user is sufficient to infer that he is either opening or closing his door.
- A smart car driving in some random place does not want a nearby fog node to know its identity when it asks for some information while in the road.

B. Privacy Threat Due to the Proximity of Fog Nodes

Any computational process or data transfer operation conducted at a fog node infers that the client is physically nearby. This issue poses a serious threat to the privacy of end-user and it also puts some limitations on the usage of existing methods [5]. For example, in a normal protocol a user can send his data to the cloud anonymously using a proxy server, VPN, or onion routing. In the case of fog computing, however, the user sends his data to the nearest fog node disclosing his approximate location to the authority controlling the fog system even if he hides his true identity. The user can possibly be de-anonymized by using some extra information about the area he lives in or using the uniqueness of the type of his data. For example, if there is a prior information that there is only one factory located in an area and is using a unique type of data, it is easy to infer all its usage information if it sends aggregate information to the cloud by comparing the same unique aggregate (say avg.) data of that area with those of other areas without such a factory.

The challenge that is related to this issue is to **prevent the nodes in upper layers from inferring information about the end-users (or devices) using the location of the fog node**.

A possible scenario showing the need for a solution for this challenge is collecting aggregate data in smart cities: Assume that there are three different neighborhoods N_1, N_2 and N_3 with three corresponding fog nodes O_{N1}, O_{N2} and O_{N3} collecting aggregate information of type g from those neighborhoods. Given a prior information that there is a facility in the neighborhood N_1 that produces data of type g, even if a homomorphic encryption scheme is used the cloud can still infer some information on the facility's usage of data type g by comparing the aggregate results of N_1 with N_2 and N_3. Note that here comparing does not necessarily mean that the cloud can decrypt the aggregate information. The comparison can be done by adding the encrypted aggregate results one by one and observing the final aggregate result. The link between the aggregate information and the facility here is the fog node since the cloud knows the actual location of the fog nodes and it knows that the facility is nearby.

V. Proposed Privacy-Preserving Paradigm

In this section, we propose a new privacy-preserving fog computing paradigm that targets the challenges described in the previous section. The new paradigm introduces two new elements to the fog architecture: The first element is a trusted third party (TTP) node distributed next to each fog node. The second element is an anonymity system between the fog-controlled layers, i.e., between the fog nodes layer and each of the upper layers.

In this work, we propose an instance for each of the generally defined elements of the new fog paradigm. As an instance of TTP we propose a TTP network constitutes of a main TTP authority and TTP fog nodes distributed next to each normal fog node. Note that other possible instances of TTP nodes can be used such as another owned computer or a trusted nearby friend. In Section V-A we propose a privacy-preserving protocol based on this instance that aims at hiding the identity of the end-user from the fog.

In Section V-B, we choose to use Mixes as an instance of the anonymity systems element and we propose a privacy-preserving protocol that aims at protecting the data of the end-user from being inferred by the upper layers of the fog architecture using his proximity to the fog nodes. Needless to say, other types of anonymity systems can be used instead such as onion routing [11].

A. Trusted Third Party Fog Nodes

In order to target the first challenge described in Section IV-A, we propose to distribute the TTP next to the fog nodes. That is, instead of one central trusted third party (TTP), there are several TTP nodes distributed next to normal fog nodes in a one to one correspondence, i.e., there is a TTP node next to each fog node. Fig. 2 shows the fog paradigm after adding the TTP nodes system.

1) Properties of the TTP node: The TTP node can be trusted for holding the true identity and the pseudonym of the IoT device. However, since the purpose of this solution is to hide the connection between the identity of the user and his private data (e.g., usage information), the TTP node should not know the data being sent to the fog node and the response of the fog node.

2) Description of the protocol: To describe how the protocol works with the new architecture, we first define some notations to be used in the protocol. Let $\mathcal{D}$ be the data sent from the IoT device to the fog node, $resp$ be the response from the fog node to the IoT device, $\mathcal{F}$ be the pseudonym (fake ID), $\mathcal{R}$ be the real ID of the IoT device, and r be a random number. $\mathcal{F}$, r and $\mathcal{R}$ are usually sent as a table consisting of multiple triplets as: $\{[\mathcal{F}_1, r_1, \mathcal{R}]...[\mathcal{F}_n, r_n, \mathcal{R}]\}$. The random number is used to distinguish the function required from the fog node. For instance, a user can use one pseudonym $\mathcal{F}_i$ with different

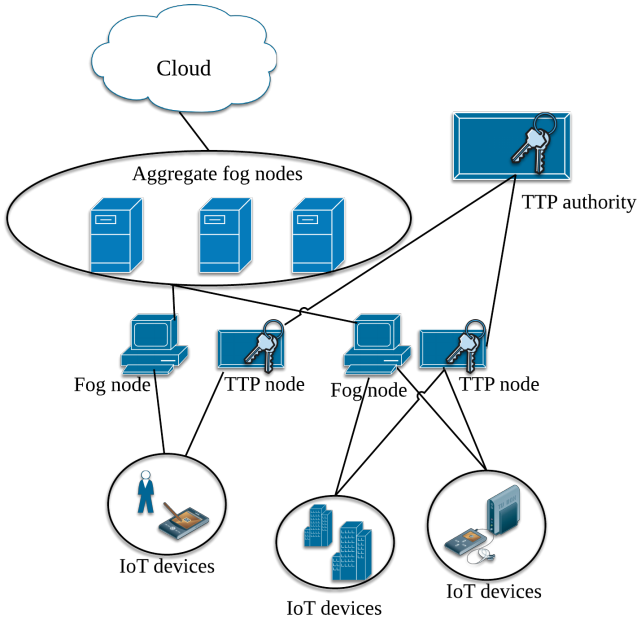

Fig. 2. Adding TTP nodes to the fog architecture.

random numbers r_l, r_m, r_n to perform three different functions on the fog node before switching to the next pseudonym. In this protocol, as well as in Fig. 3, the notation $E(x)$ means the encryption of a message x using the receiving party's public key (using asymmetric encryption). Furthermore, the subscripts used with the aforementioned notations are meant to specify an entry. For example, $\mathcal{F}$ means and arbitrary pseudonym while $\mathcal{F}_i$ means the i^{th} pseudonym in the table. The protocol flows as follows:

(i) IoT device sends a table consisting of $[\mathcal{F}, r, \mathcal{R}]$ entries to the TTP node.
(ii) IoT device randomly chooses one row from the table and sends $[E(\mathcal{D}), \mathcal{F}_i, r_i]$ as well as the address of the TTP node to the fog node.
(iii) The fog node processes the data and return $[E(resp), \mathcal{F}_i, r_i]$ to the TTP node.
(iv) The TTP node looks up which real id the response corresponds to according to $\mathcal{F}_i$ and r_i and sends the $E(resp)$ to the IoT device.

Note that the table of $[\mathcal{F}, r, \mathcal{R}]$ entries can be encrypted as well to protect the disclosure of this table by an external attacker. Fig. 3 demonstrates the flow of the protocol using the added TTP fog node.

One final remark about this protocol is about the authentication of the user. One might ask how the fog node knows that an IoT device is eligible to perform some tasks on the fog node. This issue can be solved by using anonymous credentials [12] or blind signatures [13], where the TTP node is the authority that signs and confirms the signatures.

3) Security and efficiency assessment: We assess the security of this protocol by measuring how much the fog node and the TTP node can know about the data and identity of the IoT device more than they supposed to know. From the fog node's view, it only observes a pair of pseudonym and a random number. As we assume that the fog node does not collude with the TTP node, it is hard to link $[\mathcal{F}_i, r_i]$ and $[\mathcal{F}_j, r_j]$ to the same user as the pairs generated to be computationally indistinguishable. From the TTP node's view, it knows the real identity of the user and all pairs of pseudonyms and random numbers he uses, however it is not possible to infer anything about the data being transmitted since the user does not involve the TTP node while sending the encrypted data to the fog node, also the response of the fog node is encrypted (with the public key of the user) when it goes through the TTP node.

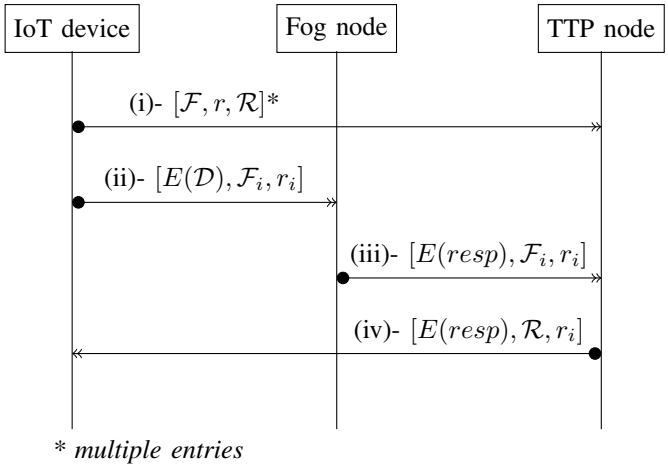

Fig. 3. TTP node protocol.

The $[\mathcal{F}_i, r_i]$ pairs can be used to further improve protecting usage patterns as follows: a user can associate a set of functions -by using different r values- with one pseudonym and another set of functions with another pseudonym. Then he can re-permute the functions randomly to different pseudonyms. In this way, the fog node cannot link between a set of pseudonyms and the usage of a set of functions.

This protocol is quite efficient since it only involves computations on the fog node and communication with the TTP nodes at the edge of the network that is as close as possible to the end-user.

B. Anonymity System between Fog Layers: Using Mixes

In order to prevent the nodes in upper layers from inferring information about the IoT devices using the location of the fog nodes, we propose to use anonymity system between the fog nodes layer and upper layers of the fog architecture. As a special case of anonymity systems, we propose to use Mixes in a mix-based protocol.

First introduced by Chaum in 1981 [14], mixes are devices used to achieve anonymity in communication between two parties. Usually a mix takes a message from party A, removes the identity of the sender, and puts it in a batch waiting for other messages to pile up. Then, evacuates them at once so that an adversary cannot link an incoming message with an outgoing message.

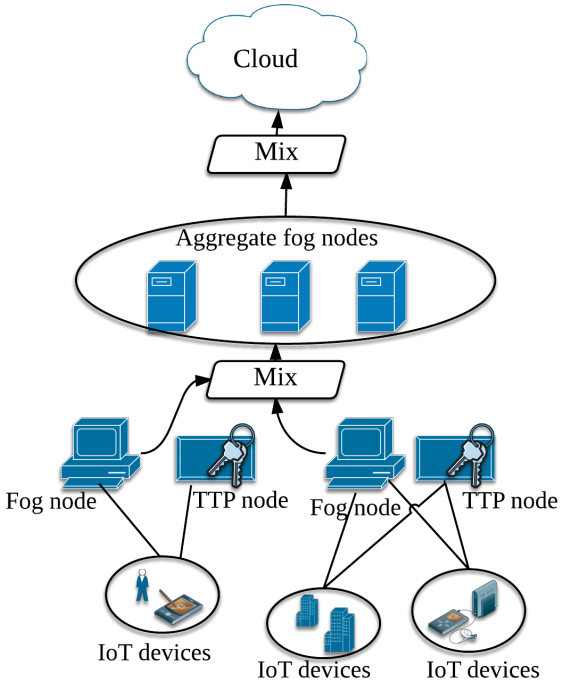

Fig. 4. Adding mixes between the fog layers.

1) Description of the protocol: The main purpose of our mix-based protocol is to hide from the cloud the address of the source fog node of a message. Fig. 4 shows the fog architecture after adding the mixes. In the following protocol, we assume that the user sends his data to the cloud encrypted using a homomorphic encryption scheme (e.g., Paillier cryptosystem [15]), and he can choose to use the mixes or not by setting a special bit to 1 (mix-bit). Needless to say, the following protocol is applied only if the user chooses to use the mixes. Furthermore, we refer to the encrypted data (with the cloud's public key) using homomorphic encryption as "ciphertext". The mix-based protocol can be described as follows:

(i) The user i sends his ciphertext c_i to the TTP node.
(ii) The TTP node checks if the mix-bit is set by the user[1] and encrypts the c_i by the mix's public key P_m and sends $E_{P_m}(c_i)$ to the fog node.
(iii) After collecting some amount of encrypted ciphertexts, the fog node sends them to the cloud through the mixes. We refer to a set of encrypted ciphertexts sent by the fog node x as T_x, i.e., $T_x = \{E_{P_m}(c_1), .., E_{P_m}(c_n)\}$.
(iv) A mix receives the set sent by the fog node and decrypts all of the ciphertexts inside it using its private key.
(v) The mix removes the sources of the messages arrived and adds them to a patch making sure that a patch is filled by entries coming from several fog nodes.
(vi) When the patch is full, the mix flushes all of the messages, $\mathcal{C}^*$, to the cloud or to the next level of the fog architecture.

[1]The mix-bit is not encrypted by the user

Fig. 5 shows the flow of the mix-based protocol. In the figure, $E(c_j)$ and $E(c_k)$ represent the encrypted ciphertexts coming from other end-users, either through the same TTP node or not. T_y and T_z represent the sets of encrypted cihpertexts coming to the mix from different fog nodes.

Note that in this protocol, in order to reduce the overhead on other types of communications between the cloud and the fog nodes, the usage of the mix is optional and it can be bypassed if the user unsets the mix-bit (in which case the mix's usage is unnecessary). Hence, the mixes are not considered a bottleneck for other communications between the fog nodes and the cloud.

2) Security and efficiency assessment: The encryption of the messages sent by the end-user with the mix's public key prevents the cloud from bypassing the usage of the mixes since it will not be able to use any information in these messages. The encryption of the messages sent by the end-user with the cloud's public key prevents the mix from inferring any information from these messages. The known attacks on mixes [16] such as the (n-1) attack [17] can be addressed using existing methods, such as Red-Green-Blue mixes [18].

Regarding the efficiency of the protocol, we assume that our solution is introduced for scenarios that do not require time-critical computations such as collecting aggregate data from smart cities. Mixes are usually considered as high-latency anonymity systems. If a lower latency solution is needed, it is possible to replace the functionality of the mixes by another low-latency anonymity system such as onion routing [11]. If onion routing to be used in place of the mixes, the only difference in the protocol is that instead of encrypting the ciphertext c_i with the mix's public key, the TTP node encrypts it with the public keys of the onion routers.

VI. Discussion

In this section, we discuss our proposed solutions in terms of their impact on existing privacy issues and their applicability and feasibility.

A. Impact on Existing Privacy Issues

1) Data privacy & usage patterns: Using the TTP fog node protocol, data and usage privacy is guaranteed since the data is encrypted throughout the protocol, and fog nodes can observe the encrypted data but they cannot relate it with end-users. Furthermore, the generation of multiple pseudonyms provides unlinkability between the end-users and the generated data through the protocol. A possible solution for protecting usage privacy from an adversary is by creating dummy tasks and offloading them to the nearest fog nodes from time to time. In this way, the adversary gets wrong usage patterns and cannot infer anything.

2) Location privacy: In the TTP fog node protocol, location of the end-users is hidden by using identity obfuscation. Even though the fog node knows the end-user is somewhere nearby, it cannot find out its correct location since it does not know its true identity.

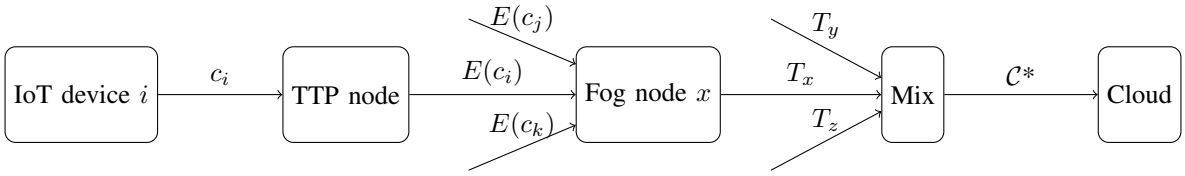

Fig. 5. The flow of the mix-based protocol

In the mix-based protocol, the location of the end-user is protected by using mixes to remove the address of the source fog node and mix all the messages so that no inference between an incoming message and outgoing message can be constructed.

B. Applicability and Feasibility

At first glance, the concept of adding a TTP node next to each fog node for only achieving privacy seems a bit costly and impractical. However, this opinion is debatable as after the adaptation of fog computing many users will need privacy-preserving solution and many companies will be able and welling to offer such a service.

Furthermore, the concept of TTP node does not necessarily mean that it has to be an external authority dedicated to provide privacy services. Rather, it can mean any third party that you trust such as another device you own or a friend who is also close to the same fog node.

The TTP node protocol provides semi-anonymity for the end-users by hiding the identity of the user from the fog node but not from the TTP node. As previously discussed in Section V-A, the TTP node can not infer any information about the data being sent/received between the fog node and the end-user. This means that the TTP node does not benefit from knowing the real and fake identity of the end-users because: it is not involved in one part of the communication ((ii) in Fig. 3), and it does not know anything about the functionality being executed on the fog node.

The mixes in the mix-based protocol seems to be a bottle-neck on the communication between the fog nodes and upper layers. However, as discussed in Section V-B the usage of the mixes is not obligatory and they can be bypassed if the user does not wish to involve them in the communication. Indeed, the usage of mixes is required for only few types of communication, specifically those which require protecting privacy. In the mix-based protocol, using the conventional homomorphic encryption schemes might cause a computational overhead on the mobile and IoT devices. To overcome this issue, we propose to use a lightweight homomorphic encryption scheme such as [19].

Both protocols are considered distributed-friendly. There can be multiple mixes for each group of fog nodes. Similarly, there can be multiple TTP authorities controlling nearby TTP nodes.

VII. Related work

To our knowledge, no previous work has defined or attempted to solve the challenges defined in this paper under the fog computing architecture. However, there are plenty of works related to protecting the privacy in smart grids by hiding the identity of the user or/and protecting the data from being disclosed or inferred by an adversary.

Lu et al. [20] propose an efficient privacy-preserving data aggregation scheme for smart grids. However, their work considers the cloud as a trusted party and does not protect the end-user from the threat mentioned in Section IV-B.

Efthymiou et al. [21] attempt to protect the smart metering privacy issue by anonymizing the identity of high-frequency metering data through an escrow service. Their work, however, does not solve the challenges defined in Section IV because it requires the communications with a distant party.

Siddiqui et al. [22] explore different methods that are proposed to protect the privacy of smart grid. However, all the methods mentioned in that paper do not solve either one or both of the challenges we define under the fog computing architecture.

Differential privacy [23] is a useful method for hiding the the actual data from inference attacks on the aggregate data provided by a database. In our case, differential privacy is not applicable since each end-user reports his own data to the fog node, which makes the aggregation and sends the aggregate data to the cloud.

VIII. Conclusion

Fog computing introduces several privacy and security threats that cannot be solved using conventional and existing methods. These threats must be studied and other possible threats must be identified before the adoption of fog computing architecture on larger scales. In this work, we introduce two privacy challenges caused by the adaption of fog computing. The first challenge is that the anonymity of the end-user or IoT devices cannot be achieved using existing methods without violating the design purposes of fog computing, such as involving distant third parties or requiring more time. The second challenge is that no matter how much the end-user tries to hide his location, it can be inferred by his closeness to the fog node. We put these challenges in the context of real-life scenarios and we propose a privacy-preserving fog computing paradigm that solves these challenges.

The proposed paradigm adds two elements to the fog architecture: Anonymity system between the fog layers and a trusted third party next to each fog node.

As an instance of trusted third parties, we propose to use a distributed network of TTP nodes controlled by a TTP authority and we propose a TTP node protocol to hide the identity of the end-users from the fog nodes. We show that the protocol is secure and efficient as it does not require the involvement of a distant party.

As an instance of anonymity systems we proposed to use Mixes and we propose a mix-based protocol to protect the privacy of end-users from being violated due to the user's proximity to the fog node. We show that the protocol is secure and we discuss how we can replace the usage of mixes with another low-latency anonymity system to improve the efficiency.

REFERENCES

[1] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*, pp. 37–42, ACM, 2015.

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC '12, (New York, NY, USA), pp. 13–16, ACM, 2012.

[3] Cisco., "White paper: Fog computing and the internet of things: Extend the cloud to where the things are.," tech. rep., 2015.

[4] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency and Computation: Practice and Experience*, vol. 28, no. 10, pp. 2991–3005, 2016.

[5] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, 2016.

[6] R. Mahmud and R. Buyya, "Fog computing: A taxonomy, survey and future directions," *arXiv preprint arXiv:1611.05539*, 2016.

[7] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[8] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey.," in *WASA*, pp. 685–695, 2015.

[9] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[10] K. Lee, D. Kim, D. Ha, U. Rajput, and H. Oh, "On security and privacy issues of fog computing supported internet of things environment," in *Network of the Future (NOF), 2015 6th International Conference on the*, pp. 1–3, IEEE, 2015.

[11] D. M. Goldschlag, M. G. Reed, and P. F. Syverson, *Hiding Routing information*, pp. 137–150. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996.

[12] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[13] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*, pp. 199–203, Springer, 1983.

[14] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[15] P. Paillier *et al.*, "Public-key cryptosystems based on composite degree residuosity classes," in *Eurocrypt*, vol. 99, pp. 223–238, Springer, 1999.

[16] M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," *ACM Comput. Surv.*, vol. 42, pp. 5:1–5:35, Dec. 2009.

[17] A. Serjantov, R. Dingledine, and P. Syverson, *From a Trickle to a Flood: Active Attacks on Several Mix Types*, pp. 36–52. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[18] G. Danezis and L. Sassaman, "Heartbeat traffic to counter (n-1) attacks: Red-green-black mixes," in *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, WPES '03, (New York, NY, USA), pp. 89–93, ACM, 2003.

[19] M. R. Baharon, Q. Shi, and D. Llewellyn-Jones, "A new lightweight homomorphic encryption scheme for mobile cloud computing," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 618–625, Oct 2015.

[20] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[21] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *2010 First IEEE International Conference on Smart Grid Communications*, pp. 238–243, Oct 2010.

[22] F. Siddiqui, S. Zeadally, C. Alcaraz, and S. Galvao, "Smart grid privacy: Issues and solutions," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*, pp. 1–5, IEEE, 2012.

[23] C. Dwork, *Differential Privacy*, pp. 338–340. Boston, MA: Springer US, 2011.