Securing the Mobile Edge through Named Data Networking

Marica Amadeo*, Claudia Campolo*, Antonella Molinaro*, Cristina Rottondi†, Giacomo Verticale‡
*DIIES Department – University "Mediterranea" of Reggio Calabria – Italy
†Dalle Molle Institute for Artificial Intelligence – University of Lugano (USI)
University of Applied Science and Arts of Southern Switzerland (SUPSI) – Switzerland
‡Dept. of Electronics, Information and Bioengineering – Politecnico di Milano – Italy

***Abstract*—The continuous growth in the number and capabilities of connected Internet of Things (IoT) and consumer devices, coupled with the increasing diversification of services running over the Internet, calls for the adoption of cutting-edge technologies to exchange, store, and process the big amount of generated data. Mobile Edge Computing (MEC) and Information-centric Networking (ICN) are candidate enabling technologies to cope with the raised issues in terms of scalability, low-latency, availability, reliability, and security.**

This paper proposes a protocol for secure and privacy-friendly service provisioning at the mobile edge, which also provides a fast way to build trust between consumers (mobile/IoT devices) and providers (middle tier servers and other mobile devices). The solution builds upon ICN pillars, in particular upon the Named Data Networking (NDN) paradigm. The security properties of the designed protocol are discussed and its behavior practically shown in a reference use case.

***Index Terms*—Information-Centric Networking, Named Data Networking, Mobile Edge Computing, Internet of Things Security**

I. Introduction

Recent studies forecast that the number of wireless connected devices will increase beyond 10 billions worldwide by 2020, with Internet of Things (IoT) and consumer devices representing the biggest share [1]. This opens up unprecedented opportunities for the provisioning of new services and it is lifting tremendously the requirements in terms of cloud-based service access. The Mobile/Multiaccess Edge Computing (MEC) paradigm [2] has recently promoted the idea to offer cloud-like services (processing, storage, and data access) at the network edge with lower latency, by relying on a *middle tier* of more powerful consumer devices and servers, typically co-located with access points (APs) and base stations (BSs) between the end-devices and the remote cloud.

Despite the inherent advantages of offering services at the mobile edge, this change of paradigm raises new challenges and calls for the design of proper schemes facilitating *service naming/description*, *on-demand deployment*, and *service discovery and access* operations. Existing edge computing solutions typically provide add-on primitives and mechanisms to support the aforementioned functionalities. Moreover, they often lack adequate security and privacy mechanisms *(i)* to validate mutually the identity and trustworthiness of requesting nodes (Consumers) and nodes offering their capabilities (Providers), and *(ii)* to ensure that input parameters as well as resulting data are kept private, especially when personalized service results are requested.

In this work, we propose a solution to support *secure cloudification at the mobile edge* "by design", based on Named Data Networking (NDN) [3], one of the most prominent instantiations of the Information-Centric Networking (ICN) architecture. The expressiveness of the NDN naming scheme and its built-in Interest/Data primitives well suit the demands of a mobile edge, as argued in [4] and more recently in [5]. Indeed, thanks to the detachment from the host-centric communication model, NDN particularly fits scenarios where the identity/location of a Provider is not a priori known and where disconnections of a Provider are frequent.

The solution designed in this paper offers a simple yet powerful framework that allows mobile devices, outside their home environment, to discover local service providers (i.e., peer mobile nodes, edge network nodes) to which they can delegate tasks that cannot be performed in a stand-alone manner, either because they are too resource-intensive for the current capability of the device, or because they need collection of some phenomena/events from multiple nodes.

Our preliminary NDN-based solution proposed in [4] leverages only on standard NDN security mechanisms, which provide only data authentication. In particular, standard NDN lacks means to verify that a request for a given content (or service) is legitimate and also lacks means to provide data confidentiality. Our extended protocol provides a mechanism for *mutual* Consumer-Provider authentication, so that *(i)* a Consumer can verify that the candidate Provider is authorized by an Access Server (AS) to provide the service, and *(ii)* the Provider can verify that an AS authorized the Consumer to ask for the service. Both the service discovery and provisioning phases are based on a cryptographically confidential and authenticated packet exchange. We call the resulting protocol Authenticated Named Data Networking at the Edge (ANDNe). It is worth noting that we present our extensions to NDN security in the context on Mobile Edge Computing, but they can also be considered a starting point for a more secure NDN in any other application scenario.

The remainder of the paper is organized as follows. Section II provides background material on our preliminary work in [4] and clearly specifies the new contributions proposed in this paper. In Section III ANDNe is presented, starting with the

design of security goals and moving to the detailed protocol description and operation. In Section IV, the viability of ANDNe is debated for a reference use case, before concluding remarks in Section V.

II. Background and Motivations

A. NDN at the edge: an overview

Although initially conceived as a networking paradigm to facilitate *named* content distribution in the future Internet [3], NDN has been recently gaining momentum as an enabling technology in the IoT research arena [6], [7], [8]. In this context, NDN can also be a good candidate to easily support cloud-like services at the network edge, as presented in [4] where we first proposed the NDNe (Named Data Networking at the edge) protocol, which extends the legacy NDN semantics, based on Interest and Data packets exchange, to support provider discovery and service provisioning at the mobile edge. The NDNe naming scheme has been designed to use well-known name prefixes to identify, not only contents, but different types of services (e.g., content storage, data compression) and their features.

A Consumer asking for a service looks for a potential Provider in its neighborhood by broadcasting a modified Interest packet, called eInt-REQ, containing the name and the parameters of the requested service. Receiving nodes process the request and check if they are able to provide the named service with the specified performance parameters. If it is the case, the nodes offer themselves as Providers by replying with an eData-REP packet containing both node and service information. The Consumer selects the most convenient candidate and sends an eInt-CONF packet as a confirmation of the service acceptance to the Provider, which, in its turn, sends an eData-ACK as an acknowledgement.

After the provider selection, the service provisioning phase starts by leveraging the *legacy* NDN packet exchange. For instance, if the Consumer asks for a data processing service (e.g., video compression), the selected Provider retrieves the content from the Consumer through Interest and Data packets, and after the content is processed, the Consumer can retrieve it by issuing Interest packets towards the Provider.

NDNe has been designed as a flexible solution to discover and access different service types; this implies that customized selection criteria can be set by the Consumer for choosing the best Provider(s). The Provider selection phase can yield to the choice of a single or multiple Providers; an example of the latter case is when the Consumer asks for a complex processing service (e.g., transcoding) that needs to be offloaded to many nodes running distributed tasks in parallel [9].

B. Contributions

The reference mobile edge environment is characterized by *highly dynamic* interactions, due to the volatile and intermittent contacts among involved nodes, whose storage, processing, battery, and connectivity capabilities may largely vary, as graphically represented in Figure 1. This is the case, for instance, of a shopping mall, a leisure place, a road, a conference venue, but also emergency response environments, where *heterogeneous* nodes, often *unknown to each other*, encounter and communicate *without a preliminary set up* and without third-party involvement. In these environments, secure and trusted node interactions must be ensured so that Consumers and Providers can find each other and agree on the service to be offered. To this purpose, in this work we overhaul the basic NDNe operation, by adding specific packet fields carrying security-related information, to enable the *mutual authentication* of Consumers and Providers while preserving Consumer privacy in the data exchange.

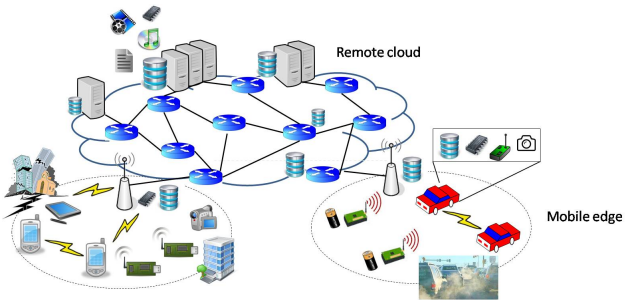

Fig. 1. Reference scenario: data processing, storage and access are available at the cloud and mobile edge levels.

In particular, the *service request*, which is part of the initial handshake in NDNe, is *cryptographically confidential and authenticated*. To achieve protection against replay attacks and forward secrecy, the Provider must maintain a list of the public keys used by the Consumers and reject requests using an old key. The Provider does not need to keep any other state after the end of a task and can delete the list when it chooses a new public key. In this way, the Provider can reuse the same public key multiple times; thus, answering a service discovery request is an idempotent operation and the response could be cached. After the initial eInt-REQ/eData-REP handshake for Provider discovery, any *content transfer* is also *cryptographically confidential and authenticated*, protected from replay attacks and enjoys perfect forward secrecy.

With respect to a pure NDN solution, our proposal protects *(i)* Consumers against a large class of attacks by dishonest intermediate nodes such as: eavesdropping of service requests, user tracking and profiling, man-in-the-middle attacks and attempts to steal credentials and *(ii)* Providers are protected against attempts to obtain service executions requested by unauthorized users, quite common in the mobile edge [10].

III. Protocol Description

A. Protocol Design Goals

In this Section we describe the security properties that we want to achieve in ANDNe. These properties must hold against any active network attacker that can intercept, modify, drop, or inject packets and can impersonate or corrupt any Consumer or Provider nodes. The attacker cannot impersonate or corrupt Consumers or Providers authorized by the Authorization Server to request or perform a given service. Authorized

Consumers and Providers are honest entities that correctly execute the protocol.

1) **Authentic Advertisements.** The Consumers should be able to detect and drop forged service advertisements. Otherwise an attacker could profile the list of services of interest to a Consumer or even trick a Consumer into sending private data to the attacker.
2) **Consumer Privacy.** An eavesdropper could determine the network interface from which a request for service S is coming but not the associated task parameters and the Consumer ID. Only the chosen Provider can learn the ID of the Consumer requesting the service.
3) **0-RTT Mutual Authentication.** If the Provider is already known from a previous discovery phase, the Consumer should be able to send authenticated input data in the first packet.
4) **Protection from Replay Attacks.** The Provider should drop any duplicate request.
5) **Protection from Denial-of-Service Attacks.** The Provider responses to service discovery requests should require minimal CPU and bandwidth usage, which is particularly relevant for potentially constrained IoT devices.

To ensure that the protocol is applicable to the context of NDN, we impose the following constraints:

1) **Compatibility with NDN/NDNe.** Nodes should communicate by leveraging NDN/NDNe primitives, without additional round trips. Intermediate nodes that are not involved in the authentication should simply forward the packets according to the usual rules.
2) **No out-of-band pairing and no dependency on external online nodes.** Consumers and Providers should be able to communicate to unknown nodes with no setup. The nodes can rely on trusted third-party nodes that can operate as registration servers and are not required to be available during the protocol exchange.

B. Attacker Model and Security Definition

Formally, we consider two major security problems: key-exchange security and private authentication. For the key-exchange problem, we assume the attacker model and security definitions in [11], which in turn is the Canetti-Krawczyk model of key exchange [12]. The attacker has full control of the network communication and can intercept, delay, drop, inject or tamper messages. This includes the ability of activate protocol entities as initiators or responders of NDN exchanges. In addition, the attacker can learn the private state of any session and corrupt any party.

Definition 1 (Key-Exchange Security): Consider an efficient attacker $\mathcal{A}$, a key exchange between an uncorrupted Consumer C and an uncorrupted Provider P, and the output of the key negotiation, atk. The following cryptographic experiment is done. At the end of the exchange, a Challenger gives the attacker $\mathcal{A}$ a key k, which can be either atk or a random key. We say that the key exchange is secure if any $\mathcal{A}$ can distinguish whether it was given atk with probability negligibly close to 1/2.

For the privacy problem, we assume Wu's attacker model and definition [11]. In this model, the adversary can passively observe any exchange and can impersonate any party except a Consumer authorized to use the service. This requirement is unavoidable, because any Provider is willing to reveal its identity to an authorized Consumer.

Definition 2 (Key-Exchange Privacy): Consider an efficient attacker $\mathcal{A}$, controlling a set of Providers $P_1, \ldots, P_n$, and a special test Consumer whose identity can be either ID_T or ID'_T. The attacker $\mathcal{A}$ can activate any Provider to generate any eData-REP or eData-ACK. However, the provider does not have a valid certificate authorizing the Provider to sign data packets with any prefix beginning with $/S/$, where S is the service requested by the text Consumer. We say that the key exchange is private if any $\mathcal{A}$ can distinguish whether the Consumer identity is ID_T or ID'_T with probability negligibly close to 1/2.

C. Protocol Script

In ANDNe the role of Consumers is played by mobile devices (e.g., smartphones, tablets, laptops) under the coverage of one or multiple edge network nodes (e.g., APs, BSs) and close to many other mobile devices. A Provider may be either a mobile device or an edge node.

We assume that the prospective Consumer and the prospective Provider have previously registered themselves with an AS and received the following public parameters and the public key certificates described below, which also serve as authorization tokens:

- The name of the desired service S and a public key pk_S used to confidentially send service parameters to any authorized Provider.
- A key generation function KDF.
- A public key encryption scheme PK.Enc.
- An Authenticated Encryption Scheme AEnc.
- A group $\mathbb{G}$ in which Hash-DH and Strong-DH assumptions hold and a generator g of the group.

The Consumer initially knows:

- Its own ID ID_C.
- A private signing key and a certificate cert_C issued by the AS that binds the Consumer ID to the signing key and authorizes the Consumer to use the service. We will indicate with $\mathsf{sig}_C(\cdot)$ the signatures generated by the Consumer.

The ith Provider initially knows:

- The private key sk_S used to decrypt service parameters. This private key is shared by all the providers.
- The Provider identifier ID_i.
- A private signing key and a certificate cert_i issued by the AS that authorizes the provider key to sign Data packets with prefixes $/S/\texttt{discovery}/$ and $/S/\mathsf{ID}_i/$. This ability also implies that the Provider is authorized to provide service S.

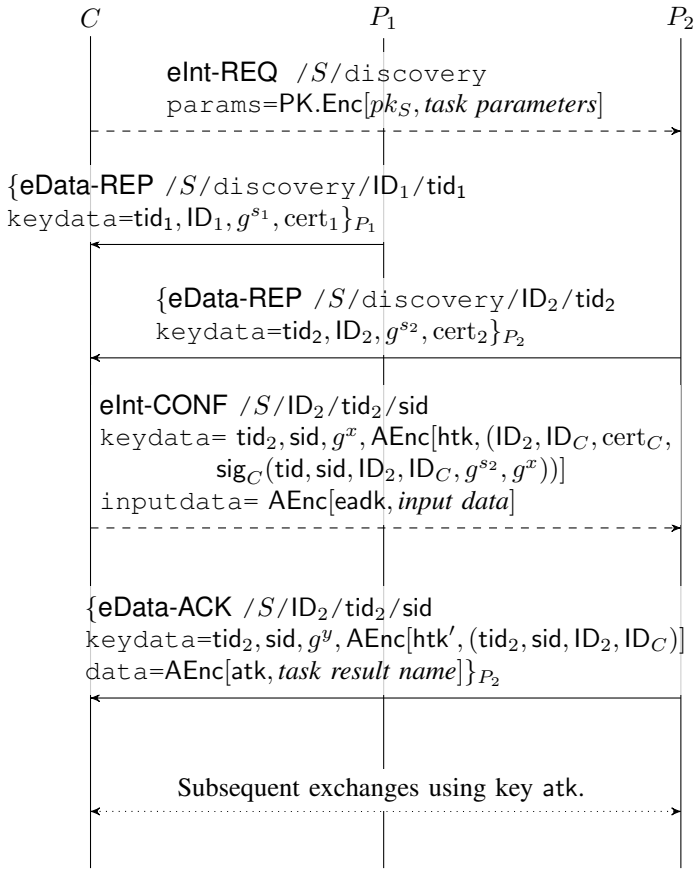

Fig. 2. Steps of the ANDNe protocol. A Consumer, C, and two Providers P_1 and P_2, are shown, which are both able to provide the service S.

In the following, the notation $\{X\}_j$ means that the payload X is sent in cleartext and also signed by node j using the usual NDN procedures.

The protocol comprises the following steps (Figure 2), which overhaul the basic interactions of NDNe with proper authentication mechanisms:

1) The Consumer encrypts the service parameters using public key encryption with public key pk_S. Encryption can be omitted if the service parameters are public. The Consumer includes the resulting ciphertext in the `params` field of the request. It then creates a content name concatenating the service name S and the keyword `/discovery` and sends an eInt-REQ packet.
2) Each Provider i receiving the eInt-REQ packet decrypts the parameter field, obtaining the service parameters. If the provider is not able to perform the service, it drops the request. Otherwise it generates a temporary ID tid_i and a temporary private key s_i. The tuple $(\mathsf{tid}_i, \mathsf{ID}_i, g^{s_i}, \mathrm{cert}_i)$ is included in the `keydata` field of the answer packet. It creates a content name concatenating the original interest, the Provider ID ID_i, and $\mathsf{tid_i}$. The answer packet is signed by P_i and is sent as an eData-REP.
3) The Consumer chooses the provider P_j among the candidate Providers and retrieves the key data. It verifies that cert_j is signed by the AS and then verifies the packet P_j signature. If any of these steps fails, the Consumer chooses another provider or stops. Otherwise, the Consumer derives the keys

$$(\mathsf{htk}, \mathsf{htk'}, \mathsf{eadk}) = \mathsf{KDF}(g^{s_j}, g^x, g^{s_j x}), \qquad (1)$$

generates a session id sid and a session private key x. Then it signs the tuple $(\mathsf{tid}_j, \mathsf{sid}, \mathsf{ID}_j, \mathsf{ID}_C, g^{s_j}, g^x)$ and encrypts the result, along with ID_j, ID_C, and cert_C using htk obtaining the string c'_3. The specification of the service to be performed, including any pointer to the input data, is encrypted using eatk obtaining the string c''_3. Finally it creates an eInt-CONF packet for the content name $/S/\mathsf{ID}_j/\mathsf{tid}/\mathsf{sid}$ and includes $\mathsf{tid_j}, \mathsf{sid}, g^x, c'_3$ in the `keydata` parameter and c''_3 in the `inputdata` parameter.
4) The Provider P_j derives the keys $(htk, htk', eadk)$ using (1) and retrieves the encrypted content. Then, it verifies the authenticity of the certificate cert_C and that the Consumer is authorized to use the service. Then, it verifies the Consumer signature. If any of these steps fails, then the Provider stops. Otherwise, it proceeds to execute the required task, including the retrieval of any required input. Then, it generates a private key y and derives the key

$$\mathsf{atk} = \mathsf{KDF}(g^{s_j}, g^x, g^{s_j x}, g^y, g^{xy}).$$

The Provider encrypts the tuple $(\mathsf{tid}_j, \mathsf{sid}, \mathsf{ID}_j, \mathsf{ID}_C)$ using $\mathsf{htk'}$ obtaining c'_4 and the service result using atk obtaining c''_4. Finally, the Provider sends an eData-ACK packet including tid, sid, g^y, c'_4 in the `keydata` field, and c''_4 in the `data` field.

D. Evaluation of Protocol Overhead

For the computation of message overhead we assume the following cryptosystems. For symmetric encryption, we consider AES with the Galois Counter Mode of operation (GCM), which adds to the message size a 128-bit nonce and a 128-bit authentication tag. Symmetric keys are generated via the Elliptic Curve Diffie-Hellman protocol (ECDH) defined over a 256-bit-sized prime field. Considering some overhead, each point is encoded as a 512-bit string. For asymmetric encryption, we adopt a hybrid scheme in which standard RSA with key length of 2048 bits is used to encrypt a randomly generated key of 128 bits. For signatures, we adopt the ECDSA algorithm with key size 256 bits, which results in an encoded signature length of 576 bits. The size of the certificates is assumed to be 8400 bits, whereas party IDs and temporary IDs are assumed to be 128 bits long. The overheads due to the cryptographic content of each message are reported in Table I. Note that the most significant contribution to the sizes of the eData-REP and eInt-CONF is the size of the included certificates.

In terms of computational complexity, the most expensive operations necessary to process an eInt-REQ message are

TABLE I
SECURITY OVERHEAD PER ANDNE MESSAGE

Message	Overhead (bits)
eInt-REQ	2304
eData-REP	9168
eInt-CONF	10256
eData-ACK	1536

a public key encryption at the Consumer side and a corresponding decryption at each Provider. These operations can be waived in case of resource constraints, at the expense of some privacy loss. Generating an eData-REP requires a point multiplication on the elliptic curve, and a signature. Since the same eData-REP can be precomputed and reused to answer multiple messages, the cost for the Provider is negligible. Instead, the Consumer must pay the verification of a packet signature. Both the eInt-CONF and the eData-ACK messages require two point multiplications and a signature generation at the transmitter side and a point multiplications and a signature verification at the receiver side. These are the two most expensive messages in the protocol. However, they are used when the service discovery phase is finished and must be paid only once for each service request. Therefore, their cost can be amortized in the cost of the service itself.

E. Security Discussion

This Section discusses how the proposed scheme provides key-exchange security and privacy according to the definitions in Section III-B. The protocol exchanges two different keys between the Consumer and the Provider. An early key eadk and a final key atk. With respect to key atk, the cryptographic part of our authentication protocol is very similar to the Private Service Discovery Protocol (PSDP) in [11], which in turn is a variation of SIGMA-I [13]. There are two differences between our protocol and PSDP: (1) in ANDNe, the provider certificate and the provider signature in the eData-REP message are not encrypted, whereas in PSDP the server's broadcast message they are encrypted and (2) in ANDNe, the eData-ACK message is signed even though it is also protected by means of the Authenticated Encryption scheme. Both these modifications have no impact on the security proof in [11, Theorem E.1]. Therefore, key atk is secure as in Definition 1.

Similarly to PSDP, the ANDNe protocol negotiates the early key eadk, which can be used to encrypt and authenticate the input data in the eInt-CONF message. This makes it possible for the Consumer to send the service parameters while the key exchange is still ongoing, thus saving at least one round-trip time in the service request. Security of the key eadk was proved in [11, Theorem E.2] using the one-pass model, which does not guarantee perfect forward secrecy and makes the Provider vulnerable to replay attacks. In fact, each Provider reuses the same private key for different exchange with different Consumers. This makes it possible to cache the eData-REP message, allowing the Consumer to avoid multiple key generations and broadcasts and allowing Consumers to send eInt-CONF messages without repeating the Service Discovery phase. The price is that an attacker could replay an old eInt-CONF resulting in a spurious Consumer authentication. To prevent these replay attacks, the Provider must keep track of the public keys advertised by the Consumers and drop any eInt-CONF packets that reuse an old key g^x until the Provider changes its own key pair. It is worth noting that the key space is large, so it is unlikely that a honest Consumer reuses a public key. If the Provider performs such bookkeeping, also the early key eadk is secure as defined in Definition 1. This is the only bookkeeping required to the Provider.

Regarding key-exchange privacy, ANDNe provides Consumer privacy. Since disclosing the Provider ID is necessary for packet routing and signing, implementing mutual privacy requires modifications to how NDN nodes manage NDNe packets, which is out of the scope of this paper. Formally, we note that privacy against passive attackers is guaranteed because the eInt-REQ message is encrypted with a semantically secure encryption scheme and the subsequent key exchange is analogous the the SIGMA-I protocol, which is private against passive attackers [13]. With respect to active attackers, the Consumer verifies any incoming eData-REP messages and drops them if they do not contain a valid certificate signed by the Authorization Server. Therefore, the Consumer never reveals its ID to an attacker-controlled Provider.

With respect to our design goals, we observe the following. Goal 1 (Authentic Advertisements) is a consequence of protocol security in the CK model. Goal 2 (Consumer Privacy) is obtained by encrypting the Consumer ID and signatures. Goal 3 (0-RTT Mutual Authentication) is a consequence of protocol security in the one-pass model. Goal 4 (Protection from Replay) is obtained with the bookkeeping described above. Finally, Goal 5 (Protection form Denial-of-Service) is guaranteed because eData-REP messages can be cached and repeated at each service discovery request.

IV. REFERENCE USE CASE

To show how the envisioned mechanism works in practice, we consider the following exemplary emergency scenario.

The town of Alice is struck by a Category 4 hurricane. Fortunately, she is safe, but her house is devastated. Fallen trees and power poles have isolated almost all the residential area. When the storm melts away, Alice makes an high quality (full HD) video of her house and the surrounding environment and tries to send it to the local Emergency Management Agency, which has prepared a crisis response service. Since the phone line and the nearby Wi-Fi hot spots are down, the authorities are arranging access points (APs) in the streets as data mules, to collect information about the damages and offer connectivity to the people.

Alice finds the emergency AP service, but realizes that the connection is slow, probably due to congestion, and a long video upload is infeasible. So, Alice decides to perform a video compression to reduce the cost of the communication. Such application is not available in her smartphone and she needs to demand the compression task to another nearby device.

ANDNe comes into the picture right now: it can arrange secure services between close devices (e.g., the smartphone of Alice neighbours), by leveraging ad hoc connectivity.

Alice's smartphone acts as an NDNe consumer C and broadcasts an eInt-REQ to discover 1-hop far away providers. The request includes: *(i)* the concatenation of service name and the keyword discovery, `/videocompression/discovery`, in the name field, *(ii)* the service parameters that specify the video size and format, the current quality in terms of definition (1080p) and the expected lower quality (360p), in the parameter field. The service parameters are encrypted with the public key common to all the authorized providers of the video compression service. This makes it impossible to an attacker to fingerprint Alice's device and track her movements.

Two authorized devices, P_1 and P_2, receive the request, decrypt the parameter field and realize they can manage the service. Both nodes defer the transmission of the eData-REP message by a time whose duration is inversely proportional to their ability in performing the service [4]. This message is not encrypted.

In our example, P_1 has more available resources than P_2 and sends the reply first. The eData-REP is signed by P_1, which has generated a temporary private key s_1, and it includes as content name the concatenation of the eInt-REQ's name, the provider ID ID_1, and a temporary ID tid_1.

C verifies that P_1 is an authorized service provider and sends an eInt-CONF packet, as a confirmation of the service request. This message also conveys the Consumer's ephemeral public key g^x, a unique identifier of the service request, sid, and the name of content that P_1 should retrieve and re-encode. These parameters are encrypted to avoid fingerprinting and tracking of Alice's device and to protect the confidentiality of the names of Alice's content. Additionally, this message is signed with Alice's key. This is different from standard NDN behavior in which Interest packets are not signed. ANDNe introduces this signature because an eInt-CONF is a service request which might be expensive to execute and for which Alice should be accountable.

Upon reception of this message, P_1 checks that the unique identifier sid and the ephemeral public key g^x are new and that Alice is authorized to request the service. If any of these checks fails, P_1 stops the protocol. Otherwise, it sends an eData-ACK as an acknowledgement and specifies the name that it will give to the processed content (to allow subsequent retrieval from C). This message is encrypted using the exchanged key atk and can be decrypted only by C.

P_1 then issues a standard NDN Interest packet for the content to be processed, and a sequence of NDN Interest and Data packets is exchanged. In this scenario, the content to be processed is provided by the same node requesting the service, but it could be any content in the network. These packets are encrypted and authenticated using key atk, except for the content name, which is necessary for routing the packets. However, this name can be generated randomly and conveys no information. As soon as P_1 begins receiving the content, it can start executing the requested task. At the expiration of the estimated processing time (as announced in the eData-REP), C starts issuing Interest packets with the name of the processed data. These packets are also encrypted and authenticated using key atk.

V. Conclusion

In this paper, we propose ANDNe, a novel protocol for *privacy-friendly service discovery* and *access* which enhances the legacy NDN Interest/Data handshake to carry-out service discovery and Provider selection procedures in edge computing scenarios, by building trust between the involved parties (i.e., heterogeneous consumer and IoT devices and network nodes at the edge). Among many possible attractive applications of the protocol, to showcase how ANDNe works in practice, we discussed the offload of a video compression task demanded from a mobile device to a nearby node in the edge under emergency conditions.

Future works will be devoted to quantitatively assess the performance of ANDNe and its effective resilience to possible attacks under a wide range of possible use-cases (spanning different service types/workloads, mobility patterns, communication technologies).

References

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 20162021 white paper," March 2017.

[2] P. Corcoran and S. K. Datta, "Mobile-edge computing and the internet of things for consumers: Extending cloud computing and services to the edge of the network," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 73–74, 2016.

[3] L. Zhang *et al.*, "Named data networking (NDN) project," Xerox Palo Alto Research Center-PARC, Tech. Rep. NDN-0001, 2010.

[4] M. Amadeo, C. Campolo, and A. Molinaro, "NDNe: Enhancing named data networking to support cloudification at the edge," *IEEE Communications Letters*, vol. 20, no. 11, 2016.

[5] D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and D. Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and research directions," in *Proceedings of the Workshop on Mobile Edge Communications*. ACM, 2017, pp. 7–12.

[6] S. K. Datta and C. Bonnet, "Integrating named data networking in internet of things architecture," in *Consumer Electronics-Taiwan (ICCE-TW), 2016 IEEE International Conf. on*, 2016, pp. 1–2.

[7] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the IoT: Experiments with ndn in the wild," in *ACM Int. Conf. on Information-centric Networking*, 2014.

[8] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: challenges and opportunities," *IEEE Network*, vol. 30, no. 2, pp. 92–100, 2016.

[9] M. Chen *et al.*, "On the computation offloading at ad hoc cloudlet: architecture and service modes," *IEEE Communications Magazine*, vol. 53, no. 6, 2015.

[10] 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*, vol. 78, pp. 680–698, 2018.

[11] D. J. Wu, A. Taly, A. Shankar, and D. Boneh, "Privacy, discovery, and authentication for the internet of things," in *ESORICS 2016: 21st European Symposium on Research in Computer Security*. Springer International Publishing, 2016, pp. 301–319.

[12] R. Canetti and H. Krawczyk, *Security Analysis of IKE's Signature-Based Key-Exchange Protocol*. Springer Berlin Heidelberg, 2002.

[13] H. Krawczyk, "Sigma: The sign-and-mac approach to authenticated diffie-hellman and its use in the IKE protocols," in *Annual International Cryptology Conference*. Springer, 2003.