Private VNFs for collaborative multi-operator service delivery: an architectural case

Gergely Biczók, Balázs Sonkoly, Nikolett Bereczky
MTA-BME Future Internet Research Group
Budapest Univ. of Technology and Economics
Email: {biczok,sonkoly,nikolett.bereczky}@tmit.bme.hu

Colin Boyd
Department of Telematics
Norwegian Univ. of Science and Technology
Email: colin.boyd@item.ntnu.no

***Abstract*—Flexible service delivery is a key requirement for 5G network architectures. This includes the support for collaborative service delivery by multiple operators, when an individual operator lacks the geographical footprint or the available network, compute or storage resources to provide the requested service to its customer. Network Function Virtualisation is a key enabler of such service delivery, as network functions (VNFs) can be outsourced to other operators. Owing to the (partial lack of) contractual relationships and co-opetition in the ecosystem, the privacy of user data, operator policy and even VNF code could be compromised. In this paper, we present a case for privacy in a VNF-enabled collaborative service delivery architecture. Specifically, we show the promise of homomorphic encryption (HE) in this context and its performance limitations through a proof of concept implementation of an image transcoder network function. Furthermore, inspired by application-specific encryption techniques, we propose a way forward for private, payload-intensive VNFs.**

I. Background and Motivation

Proposed 5G verticals are predicted to be real value-added services. A majority of these are projected to be delivered as a service chain with multiple, independent business entities contributing; either for maintaining flexibility in resource allocation or for the lack of geographical footprint. Furthermore, users (and the market as a whole) would benefit from a streamlined, one-stop shopping experience, where the customer only has to contact a single operator for setting up a complex service; who would in turn outsource and subcontract to other operators if it lacks the footprint (similar to mobile roaming), the resource capacity or can just make a larger profit. The key enablers for such collaborative and flexible multi-operator service delivery are Software Defined Networking (SDN) and Network Function Virtualisation (NFV). NFV in particular makes it convenient for an operator to outsource network functions to another operators. The Horizon 2020 5GEx project aims at creating such a service delivery platform [3].

From the technical point of view, the 5GEx architecture enables the unified cross-domain orchestration of network and cloud resources over multiple administrative and technology domains [7]. Services are realized via Service Function Chaining (SFC) potentially over domains of multiple operators. In this environment, the contract structure, the lack of trust and SFC (the path and VNFs which customer data passes through) create a complex privacy situation (adding to general NFV issues [5]; see Figure 1 for an example.

First, the user forms a trust relationship with its customer-facing operator (Op1) upon buying a service and signing a contract (Service Level Agreement, SLA). Given that Op1 potentially outsources some parts of the required service chain to other qualified operators (Op2 and Op3), user traffic will be steered through and potentially processed by VNFs in Op2's and Op3's administrative domains. Since the user does not have a trust relationship to the subcontractors Op2 and Op3, she might want to do something about the risk of Op2 or Op3 looking into her traffic. A logical step would be to apply encryption at the user side, or at the egress of Op1, before user traffic leaves the premises of the trusted domain. In this example, User 2 is the destination of User 1's traffic, therefore he is the one to decrypt it upon arrival.

Second, VNFs usually have a policy input besides the data traffic. In the given setting, it is plausible to assume that policies (firewall rules, filter expressions, coding parameters, etc.) come from Op1, while the VNF is running on the infrastructure of another operator (Op2). Now, Op1 may have a number of reasons not to expose its policies to Op2, including being competitors and hiding its cyber-defense strategies. Therefore, it could be beneficial for Op1 if encrypted policies could be interpreted by the VNF. Further complicating things, the VNF implementation could be provided by Op1, Op2 or a third-party VNF provider [2] (not depicted in Figure 1). All three cases require a different approach if Op1 wants to keep

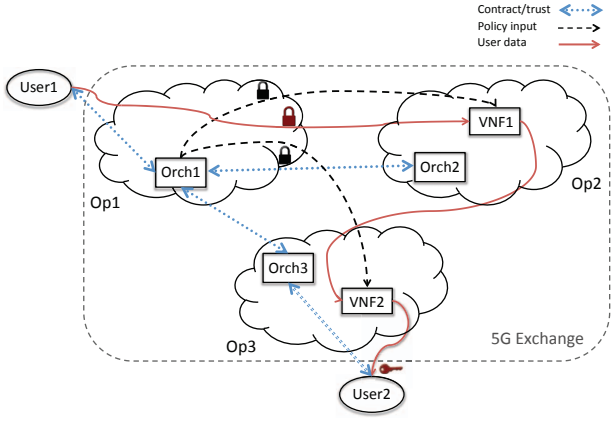

Figure 1. Multi-operator service delivery with SFC

978-1-5090-0223-8/16/$31.00

its policies, or even the function the VNF implements, hidden and at the same time, successfully outsource the operation to Op2. Even if there is a contract (and so some level of trust) between Op1 and Op2, an honest-but-curious Op2 could still pose problems to Op1.

Our contribution. In this paper, we ask the question: is it possible to remedy these situations using already developed techniques for enabling *private VNFs*? We believe that the answer is both yes and no. Yes, as homomorphic encryption seems to be a fitting solution: we show how customer data, operator policy and VNF code might be kept private to the benefit of their respective owners. We outline how homomorphic encryption (HE) [11] could be utilised to solve these issues. And no, as we implement and evaluate a proof of concept HE-enabled VNF prototype for image transcoding, and demonstrate its low performance. Building on these findings and inspired by CryptDB [16], we propose VNF class-specific encryption as a candidate solution, where HE characteristics can be satisfied via a restricted set of simpler but faster encryption schemes. Specifically, we make a case for payload-intensive VNFs, such as media transcoding. We also describe potential future work; we believe that a HE capable (whether through general HE or restricted) 5G network architecture with VNFs could solve a number of privacy issues arising in multi-operator service delivery.

The rest of the paper is organised as follows. Section II introduces homomorphic cryptography and its potential for private VNFs. Section III presents our proof of concept VNF and its performance limitations. Section IV outlines VNF class specific encryption and future work. Finally, Section V concludes the paper.

II. Homomorphic crypto: overview and applicability in the NFV context

Overview. The shift towards remote storage and processing of data has brought new demands to secure data not only while in transit or at rest, but also while it is undergoing processing. The cryptographic community has responded to the demand by developing new cryptographic tools, particularly multi-party computation, functional encryption, and homomorphic encryption. Of these, homomorphic encryption is the appropriate choice when private data is sent to the cloud for processing and should be returned to users for decrypting after processing has taken place.

While homomorphic encryption has existed for decades, for a long time the functions which could be processed were essentially limited to a single type of operation (either addition or multiplication) which severely limits the application. The conceptual breakthrough of Gentry in 2009 [11] completely changed the picture, showing that, at least in principle, *any* function can be processed on the encrypted data. This is known as fully homomorphic encryption (FHE). Gentry's original scheme, while theoretically "efficient", was hopelessly impractical due to huge public keys, ciphertexts and processing delays. Since that time there has been extensive research activity aimed at improving the efficiency of FHE while, at the same time, related encryption schemes which enable limited forms of homomorphic computation have also been realised.

Today we have a situation where modest functions can be processed homomorphically in what may be practical in some scenarios. For example, the iDASH competition winners, announced in March 2015 [1], were able to statistically process genomic data homomorphically in under a second. This shows that real applications can use the technology. However, homomorphic encryption remains orders of magnitude less efficient than ordinary asymmetric (public key) encryption, while there are theoretical grounds to believe that symmetric key homomorphic encryption can never be achieved more efficiently than asymmetric (homomorphic) encryption.

Towards a HE capable network architecture. Before deciding on how homomorphic encryption can be applied to protect VNFs, we should decide what we are trying to achieve by such a step. What should we mean by *private* VNFs? Privacy may be relevant for one or more of: the payload in packets; the function used to process the packets; and the header data.

Perhaps the most obvious aspect of privacy is to *protect the client data* while it is being processed in the network. As explained above, fulfilling this task is the aim of (fully) homomorphic encryption. In the next section, we describe a case study examining this functionality applied to image transcoding. In principle any function can be computed, but note that here we focus on functions which process user payload: functions dealing only with header data can be processed while user data is protected with conventional encryption. Putting aside the efficiency problem for now, we note that key management is a practical concern which needs to be in place with suitable key exchange between the end points before secure processing can begin. The network itself does not need any key to process the data homomorphically.

In previous work, Melis et al. [14] use the term *private NFV* to describe any scheme which protects "both the network function and its output". They achieved this by sending from the client to the cloud the description of the function encrypted homomorphically. They also implemented their scheme proving feasibility of the concept. While their work could be useful for the policy input of VNFs in our context, they have only considered simple functions which do not modify packet payload: see Section IV for payload-intensive VNFs. Moreover, while they provide a security analysis and proof, their security definitions are not as strong as would be desirable. Another potential limitation is how to decide who should choose the network function parameters. Should this be up to the client? Does it need to be authenticated? Overall there is still considerable work remaining to provide such a privacy service on a large scale.

Hiding source and destination of packets can only be achieved if we also *encrypt packet headers*. In principle this can also be done by homomorphic encryption. However, even if this can be done efficiently it is doubtful that it alone is sufficient to hide routing information without in addition using methods to combine packets from different users as is done in practical methods such as Tor [10]. Note that we do not deal with this functionality in this paper.

III. Proof of concept

Case study: image transcoding. As a first step towards the envisioned HE capable network architecture, we have implemented a proof of concept NF prototype for the first use-case. The addressed function can be considered as a part of a multi-user chat application. It is capable of sharing HE encrypted images among the users of a given group. Based on a predefined configuration parameter, a user receives an image "transcoded" to a given quality or file size. By this means, the sent picture is adjusted according to the characteristics of the user's Internet access or the contracted SLA between the user and the service provider. In our simple example, "transcoding" manifests in the rescaling/resizing of the image by averaging the RGB values of contiguous blocks. Theoretically, a similar approach can be applied to design e.g., a phone conference NF (mixing encrypted channels) or a video transcoding NF (adjusting the quality of encrypted video streams).

In our HE capable framework, images are sent in a HE encrypted format to the NF where the averaging can be performed without decrypting the content. In our current implementation, only the summation is done by the NF while the division is delegated to the hosts and performed on the decrypted content. The division has the same complexity for plain text and ciphertext, and we analyze the different operations separately. Therefore, this is not a limitation and the additional operation can easily be added to the NF later[1].

It is worth noting, that processing encrypted images is not a novel idea. For example, Intel designed [9] a coding scheme and a multi-step process including GPU based encryption at the host, image processing (resizing, cropping) at a server, and finally decoding at the host; Zhen and Huang [19] define an HE scheme which can compress preliminary encrypted pixels. However as far as we know, adding an image processing NF as a general component to an NFV framework is a new concept.

Our proof of concept prototype makes use of two open source libraries. On the one hand, OpenCV [4] (Open Source Computer Vision) is a widely used library in different fields of image processing. It includes several functions and algorithms from low-level to high-level processing tasks. On the other hand, HElib [12] is an open source library providing essential components of HE schemes. The currently available scheme is the Brakerski-Gentry-Vaikuntanathan (BGV) [8] homomorphic encryption scheme with several elements improving the efficiency. These components have been integrated with our resource orchestration framework called ESCAPE [17]. ESCAPE is a proof of concept implementation of the SFC control plane architecture proposed by the project EU FP7 UNIFY [18].

Performance measurements. We have set up a simple environment where the performance of the implemented NF can be analyzed based on different metrics. The size of the original image and the shrinking parameter are the relevant inputs. Here, image transcoding/resizing is expressed by a single shrinking parameter s defining non-overlapping $s \times s$ square blocks within the pixels, which have to be averaged.

[1]This was only an implementation issue because the used HE library does not support the required operation yet.

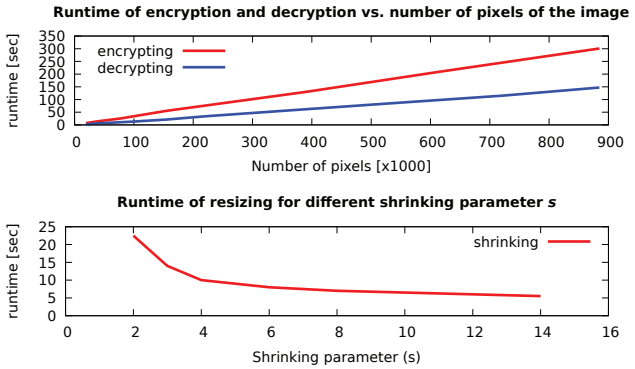

Figure 2. Run time of the algorithms

The run time of subsequent steps and the memory usage can be measured during the experiments.

The experiments were conducted on a host notebook (with 1.90 GHz CPU cores, 4GB RAM, MS Windows 8.1) running a 64 bit Ubuntu 14.04 virtual machine (with 2 cores, 2GB RAM) on top of VirtualBox 4.3.12.

The run time results for the encryption and decryption processes are shown in the upper part of Figure 2, respectively. The plots show linear scaling with the number of pixels (image size), however, the run times are extremely (unacceptably) long. It is worth noting though that these operations have to be performed by the hosts and not the NF.

The performance of the resizing process, i.e., averaging (without division), done by the NF is presented by the lower part of Figure 2. This shows an exponential dependence from the shrinking parameter. Here we do not show the run time of the division as it was performed at the host at a much lower time scale (it was always within 0.06 sec).

Besides the run times, we have analyzed the memory consumption, as well. The relevant step is the encryption which needed approximately 150 MByte RAM per 100,000 pixels. For our largest image (1152×768), this resulted in the allocation of almost 1.5 GByte memory, which can be deemed impractical.

These performance results mainly characterise the used HE library and or quite naive reference implementation. More specifically, we used the basic data structures and supported operations in a straightforward way, and have not applied any optimization hacks in the implementation. To sum up, we have showed that the concept is viable, however, there is a lot to do for constructing and implementing HE schemes, which can be used in real use-cases.

IV. Network function class specific encryption

It is clear that HE is an extremely powerful tool with a potentially wide range of applications. However, even with its ever-improving techniques and implementations, it is hard to see when it will be ready for processing large amounts of data with low delay. As payload-processing VNFs require just that, we need to find smart ways to improve processing performance.

Application-specific encryption. A promising design with potentially far-reaching impact is presented by Popa et al. [16]. CryptDB enables processing of encrypted data by most functions of SQL databases. The system builds on the fact that SQL queries are composed of a relatively small set of operations, such as equal to, less than, summing, sorting and joining. Most of these already have had efficient non-HE schemes with encrypted operation; CryptDB stitched these schemes together to cover most SQL functions. Of course, there is a performance penalty attached compared to a traditional SQL database, but authors claim the overhead being as low as 15-30%. Hence, CryptDB is fit for real-world usage for many online applications built on top of databases.

Payload-intensive VNFs. Application-specific encryption seems to suit the needs of payload-processing VNFs. As opposed to regular (header-processing) NFs, e.g., a firewall, these NFs actually modify packet payload or create new packets. First, consider a *media transcoder VNF*: in today's and tomorrow's digital content delivery, such a function can be located at a cloud provider (such as with Envivio's solution [15]) or at the Internet access provider of a user of a video streaming service, adjusting the quality of the video stream based on user device or subscription bandwidth. Second, consider the newly emerging paradigm Mobile Edge Computing (MEC) [6]. MEC offers compute and storage capabilities (as VNFs) at the edge of a mobile network, thereby offloading user equipment and enabling computationally intensive, latency-critical and location-aware mobile applications. Both for resource availability and footprint reasons, collaborative service delivery is a viable solution for mobile operators in this context. Potential MEC service scenarios include assistance in sensor data aggregation and processing, autonomous vehicle movement calculation and aggregation and video stream analytics. Although completely different on the surface, media transcoding and the various MEC-related functions rely on similar basic building blocks: matrix operations. Thus, if we construct an efficient encrypted computation method for matrix operations, we could potentially use it for a wide range of HE-emulating private VNF implementations.

Future work. The lion's share of required work lies ahead of us. In the cryptographic context, we plan to investigate application-specific encryption, first with regard to matrix operations. We would also like to look into HE implementations, and improve their performance via multi-threading and advanced data structures. Moreover, a practical key management scheme is needed for a real deployment. In the networking context, we plan to look into more advanced capabilities like migrating VNFs on-demand, where the statefulness of some VNFs could introduce further challenges. The design of contracts among the various stakeholders of the future 5G architecture (and NFV-enabled network architectures in general) should also be emphasised. One particular area of interest is security SLAs, which could be a game changer with regard to trust relationships [13].

V. Conclusion

In this paper we have outlined an architectural case for private VNFs in a multi-operator service delivery environment. We have shown that homomorphic encryption shows some promise; however, we demonstrated through a case study of image transcoding that its performance is not up to real-world use-cases in the general case. Finally, we have discussed the potential of application-specific encryption with regard to the class of payload-intensive VNFs. We believe that this research topic deserves the attention of both the networking and cryptography communities.

Acknowledgements

This work has been partly performed in the framework of the H2020-ICT-2014 project 5GEx (Grant Agreement no. 671636), which is partially funded by the European Commission. Gergely Biczók has been supported by the János Bolyai Research Scholasrship of the Hungarian Academy of Sciences.

References

[1] iDASH Agenda. http:/www.humangenomeprivacy.org/2015/AGENDA.html.
[2] Project FP7 T-NOVA. http://www.t-nova.eu.
[3] Project H2020 5GPPP 5G Exchange. https://www.5gex.eu.
[4] OpenCV: Open Source Computer Vision. http://opencv.org/, 2012.
[5] ETSI GS NFV-SEC 001 V1. 1.1 (2014-12). Network Functions Virtualisation; NFV Security; Problem Statement. 2014.
[6] ETSI GS MEC-IEG 004 V1.1.1 (2015-11). Mobile-Edge Computing (MEC); Service Scenarios. 2015.
[7] Bernardos, Carlos J. et al. 5G Exchange (5GEx) – Multi-domain Orchestration for Software Defined Infrastructures. In *EUCNC*, 2015.
[8] Z. Brakerski, V. Vaikuntanathan, and C. Gentry. Fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science (ITCS)*, 2012.
[9] C. Demerjian. Intel lets you manipulate encrypted data. https://semiaccurate.com/2012/06/27/intel-lets-you-manipulate-encrypted-data/, 2012.
[10] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: The second-generation onion router. In M. Blaze, editor, *Proceedings of the 13th USENIX Security Symposium*, pages 303–320. USENIX, 2004.
[11] C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 169–178. ACM, 2009.
[12] S. Halevi and V. Shoup. Design and Implementation of a Homomophic-Encryption Library. http://people.csail.mit.edu/shaih/pubs/he-library.pdf, 2013.
[13] M. G. Jaatun, K. Bernsmed, and A. Undheim. Security SLAs–An Idea Whose Time Has Come? In *Multidisciplinary Research and Practice for Information Systems*, pages 123–130. Springer, 2012.
[14] L. Melis, H. J. Asghar, E. D. Cristofaro, and M. A. Kâafar. Private processing of outsourced network functions: Feasibility and constructions. *IACR Cryptology ePrint Archive*, 2015:949, 2015.
[15] E. Nuage. SaaS Video Solution. http://www.envivio.com/cloud/.
[16] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 85–100. ACM, 2011.
[17] B. Sonkoly, J. Czentye, R. Szabó, D. Jocha, J. Elek, S. Sahhaf, W. Tavernier, and F. Risso. Multi-domain service orchestration over networks and clouds: A unified approach. In *ACM SIGCOMM (DEMO)*, 2015.
[18] B. Sonkoly, R. Szabó, D. Jocha, J. Czentye, M. Kind, and F.-J. Westphal. Unifying cloud and carrier network resources: An architectural view. In *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, 2015.
[19] P. Zhen and J. Huang. An efficient image homomorphic encryption scheme with small ciphertext expansion. In *Proceedings of the 21st ACM International Conference on Multimedia*, MM '13, pages 803–812, New York, NY, USA, 2013. ACM.