Managing AAA in NFV/SDN-enabled IoT scenarios

Alejandro Molina Zarca*, Dan Garcia-Carrillo*, Jorge Bernal Bernabe*, Jordi Ortiz*,
Rafael Marin-Perez† and Antonio Skarmeta*

*Department of Information and Communications Engineering
University of Murcia, Spain
{alejandro.mzarca, dan.garcia, jorgebernal, jordi.ortiz, skarmeta}@um.es

†ODIN Solutions
Murcia, Spain
rmarin@odins.es

***Abstract*—This paper proposes a novel policy-based framework to manage Authentication, Authorization and Accounting (AAA) and Channel Protection security functions in IoT networks enabled with Software defined Networks (SDN) and Network Function Virtualization (NFV) technologies. The virtual AAA, including network authenticators, are deployed as VNF dynamically at the edge, facilitating the devices' bootstrapping and ruling the access control of IoT devices to the network. The enforcement of network authorization decisions in the virtual switches is carried out through SDN. Moreover, the proposed softwarized and centralized channel protection management solution allows distributing dynamically the necessary crypto-keys for IoT M2M communications, in order to establishing DTLs tunnels among IoT devices, whenever demanded by the cybersecurity framework.**

I. INTRODUCTION

Edge and fog technologies shift centralised clouds towards the edge with the aim to deliver better throughput, enable enhanced context-specific functionality, and support diverse kinds of communications. They also allow for localized functions, such as processing the security in machine-to-machine (M2M) communication required in IoT, by exploiting nearby resources. The Fog includes another infrastructural level between edge and cloud in which security functions for IoT devices can be offloaded to their vicinity.

Fog and IoT can drastically improve network connectivity at the edge by leveraging NFV and SDN. NFV presents remarkable advantages with respect to the hosting in the edge and remote cloud data centers. Dynamic provisioning of virtual security functions towards the edge of the network can enhance scalability, necessary to deal with the huge IoT traffic.

In that sense, Authentication, Authorization and Accounting (AAA) as well as Channel-Protection Network Security Functions (NSF) can be timely and dynamically deployed at the edge in virtualized and softwarized fog entities, such as cloudlets, in order to rule the security in IoT networks. To this aim, new context-aware holistic security frameworks are needed to allow orchestrating NFV managers, SDN controllers and IoT controllers, thereby providing security chaining, as well as dynamic reconfiguration and adaptation of the virtual security appliances.

In addition, there is an strong need to define proper, interoperable and highly-expressive security policy languages and models to empower users and administrators to manage, in a high-level fashion, the overall security and privacy aspects of their Fog-IoT entities across the whole ecosystem. Those policy models could serve as input for the framework orchestrators to organize and choreograph the aforementioned security services. Some security policy models [1] and frameworks [2], [3] had proposed in the past solutions to manage distributed systems. However, they are not tailored to manage cybersecurity in IoT networks and Mobile Edge Computing scenarios, as presented in this paper.

On the other hand, AAA and Channel protection NSFs have been already successfully studied and addressed in IoT networks []. However, those NSFs have not yet properly studied and exploited NFV/SDN-enabled IoT networks, where cyber-situational and policy-based security frameworks can dynamically react and mitigate cyber-attacks by deploying timely and wisely, in the proper location, the suitable vNSF.

To fill this gap, this paper proposes a novel policy-aware approach to manage AAA and channel protection in SDN/NFV-enabled IoT networks. In our proposal, the vAAA NSF, including network authenticators, are deployed and activated dynamically at the edge, facilitating the devices' bootstrapping and ruling the access control of IoT devices to the network, by relying on SDN to enforce the network authorization decisions in the switches. Likewise, the proposed channel protection management allows provisioning dynamically the necessary crypto-keys for IoT M2M communications, establishing DTLs tunnels among IoT devices.

The rest of the paper is organized as follows. Section II analyzes current state-of-the-art about security solutions for IoT systems based on NFV/SDN. Section III overviews the cyber-security and policy-based framework. Section IV presents the proposed vAAA NSF. Section V is devoted to the softwarized Channel Protection proposal. In Section VI a promising use case is presented to assess the introduced security features. Conclusions and ongoing research are drawn in Section VII.

II. Related Work

Large scale IoT deployments are comprised of disparate devices with different protocol stacks. Providing a interoperable and open bootstrapping solution will ease the deployment of the different devices of an IoT network. In this sense, to the best of our knowledge, this work is the first attempt to integrate NFV in management of IoT bootstrapping in large deployments with AAA federation support, that is compatible with diverse bootstrapping solutions.

In IoT there are different protocols to secure the communications. The Zigbee IP [4] standard, is one of the first complete solutions for IoT. It uses PANA and EAP for network access authentication. However, AAA is not considered in the standard. Currently, there is work in standardization organizations such as the IETF to define new protocols for channel protection and key exchange and distribution in IoT, such as the OSCORE[5] and EDHOC[6] protocols; the former is used to secure the communications end-to-end, while the later generates the necessary key material. CoAP documentation defines DTLS as its secure communications mechanism.

The SDN has demonstrated to be a flexible and powerful enabler to new network solutionsThe centralized control provides complete network information, therefore enhancing control decisions. SDN based solutions endows the architecture with desirable features such as flexibility, dynamism, centralized management and scalability. Current works like [7] show how the SDN can be addressed in order to mitigate security issues at different layers. Softwarization plays a key role providing with the desired scalability level to the proposal. In this sense, other researchers have presented [8] a general security proposal to manage IPSec Security Associations (SAs) in SDN networks and enabling end-to-end channel protection.

NFV technologies avoid the deployment of specific hardware equipments through the use of virtual machines running specific network functions on commodity servers. NFV provides among others, flexible provisioning, deployment and centralized management. The possibility to employ virtual network functions (VNF) to deploy security appliances is an interesting alternative to enhance an architecture with adaptive and reactive security capabilities. In this sense, in [9] authors highlights different benefits of integrating SDN within the NFV, coming up with a software-defined NFV architecture, which allows to take advantage of both, softwarization and virtualization.

Furthermore, NFV enables on-demand deployment of virtual in-network security functions, thus avoiding trac rerouting compared to classic cloud-based approaches. To this aim, in [10] an approach towards the adoption of security policies management with dynamic network virtualization is proposed. However, the joint use of SDN and NFV security features is currently at a preliminary stage and significant efforts are still required to fully exploit their benefits. Furthermore, the integration with existing security solutions, especially for IoT, is still missing.

III. Framework Overview

The ANASTACIA framework [11] [12] provides a context-aware autonomous security orchestration in SDN/NFV-enabled Fog and IoT. The framework orchestrates dynamically the security of the network according to the context obtained from agents, mitigating and countering cybersecurity threats at the edge of the network in IoT scenarios, by deploying and orchestrating Virtual Security Functions and services even over constrained IoT devices. The security framework is endowed with monitoring and reaction tools as well as innovative algorithms and techniques for threat analysis, and correlation from different sources. Thereby, increasing the overall security, including self-repair, self-healing and self-protection capabilities, not only at the core, but also at the edge of the network.

Through the use of networking technologies such as SDN-NFV and intelligent and dynamic security policy enforcement and monitoring methodologies, different virtual security appliances such as vFirewall, vIDS, vAAA, vSwitch/Router, vHoneynet, vVPN are orchestrated dynamically at the network edge.

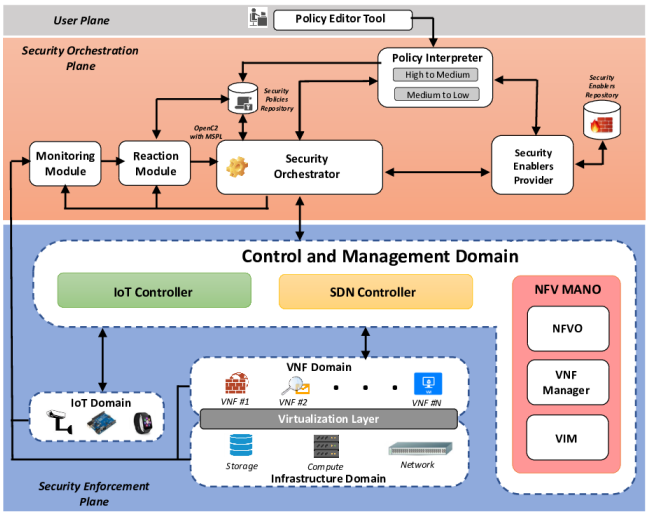

Figure 1. Anastacia Framework Architecture.

A high level view of the framework is depicted in figure III. The **User Plane** includes interfaces, services, and tools to end-users for policy definition, system monitoring and service management. Its policy editor provides an intuitive and user-friendly tool to configure security policies governing the configuration of the system and network, such as authentication, authorization, filtering, channel protection, and forwarding.

The **Security Orchestration plane** enforces policy-based security mechanisms and provides run-time reconfiguration and adaptation of security enablers, thereby providing the framework with intelligent and dynamic behavior. It is an innovative layer of our architecture and provides self-protection and self-healing capabilities for softwarized networks through novel modules.The *Policy Interpreter* module receives as input the policies and identifies the capabilities needed to enforce such policies (capability matching). Then, the Interpreter interacts with the *Security Enablers Provider* to identify the SDN/NFV-based/IoT specific enablers that are able to enforce the desired capabilities. The *Security Orchestrator* selects the enablers to be effectively deployed, accounting for the security requirements, the available resources in the underlying infrastructure, and optimization criteria. The *Monitoring* component collects security-focused real-time information related to the system behavior from physical/virtual appliances. Its main objective is to provide alerts for the reaction module in case something is misbehaving. Security probes are deployed in the infrastructure domain to support the monitoring services. Then, the *Reaction* component is in charge of providing appropriate countermeasures, by dynamically defining reconfiguration of the security enablers according to the circumstances. The reaction outcomes are then analyzed by the Security Orchestrator, which enforce the corresponding enablers' countermeasures.

The **Control and management domain** modules supervise the usage of resources and run-time operations of security enablers deployed over software-based and IoT networks. A set of distributed SDN controllers takes charge of communicating with the SDN-based network elements to manage connectivity in the underneath virtual and physical infrastructure. NFV ETSI MANO-compliant modules support secure placement and management of virtual security functions over the virtualized infrastructure. The Iot Controller manage IoT devices and low power and lossy networks (LoWPANs).

Infrastructure and Virtualization domain This domain comprises all the physical machines capable of providing computing, storage, and networking capabilities to build an Infrastructures as a Service (IaaS) layer by leveraging appropriate virtualization technologies. This plane also includes the network elements responsible for traffic forwarding, following the rules of SDN controllers, and a distributed set of security probes for data collection to support the monitoring services. The **VNF domain** accounts for the VNFs deployed over the virtualization infrastructure to enforce security within network services. It provides advanced security VNFs (such as virtual vFirewall, Intrusion Detection (vIDS), vAAA, channel protection, etc.),capable to provide the defense mechanisms and threat countermeasures requested by security policies. The **IoT domain** comprises the IoT devices to be controlled. This includes the security enablers, actuators or software agents needed to enforce the security directives coming from the orchestration plane and managed, at the enforcement plane, by the IoT controller.

IV. vAAA in SDN/NFV enabled IoT networks

Part of the innovations in this work is to dynamize the deployment including the core infrastructure (AAA) through the use of NFV MANO establishing the necesary certificates shared between the different partners of the federation.

A. AAA preliminaries

The Authentication, Authorization and Accounting (AAA) framework is used and instantiated, typically, in protocols such as RADIUS [13] and Diameter[14] that give support to a great number of devices. Examples of this are the Eduroam network, or TELCOS mobile deployments. They are used to authenticate the devices, authorize access to the services offered (e.g. Access to the Internet) and keep track of the use. Advanced features such as federation (e.g. exemplified in Eduroam) bring scalability to the deployment of a great number of devices that may belong to different organizations under deployment infrastructures of different operators. The Extensible Authentication Protocol (EAP) is a protocol that offers a myriad of authentication methods, as well as a Key Management Framework (EAP-KMF [15]) that enables the bootstrapping of different Unicast or Multicast security association protocols (e.g. DTLS) to secure the communications. EAP lower layers, such as PANA or CoAP-EAP transport EAP between a device and the domain controller to authenticate and provide access to the different services of the domain.

B. Bootstrapping

IoT brings heterogeneity of devices and radio technologies, with different capabilities and requirements, that have to cooperate and coexist. In ANASTACIA, we provide a VNF called vBootstrapping, that deploys an EAP lower layer (PANA [16] or CoAP-EAP [17] depending on the requirements) for IoT device bootstrapping, authenticate them and manage network access authentication. Having a VNF that deploys an EAP lower layer capable of adapting to each deployment needs, provides the flexibility needed to enable deployments to scale.

Although bootstrapping encompasses several aspects, for the sake of simplicity, this paper focuses mainly on the EAP authentication to provide network access, key derivation and distribution to bootstrap other protocols.

The proposed bootstrapping process is shown in Figure 2. Firstly, the SDN controller receives from a vSwitch a petition from an IoT device to authenticate (either PANA or CoAP-EAP) and deploys (or redirects the traffic to an already deployed VNF with the EAP lower layer in question). After the EAP authentication (step 1) is completed, the EAP lower layer has communicated with the AAA server (step 2) that

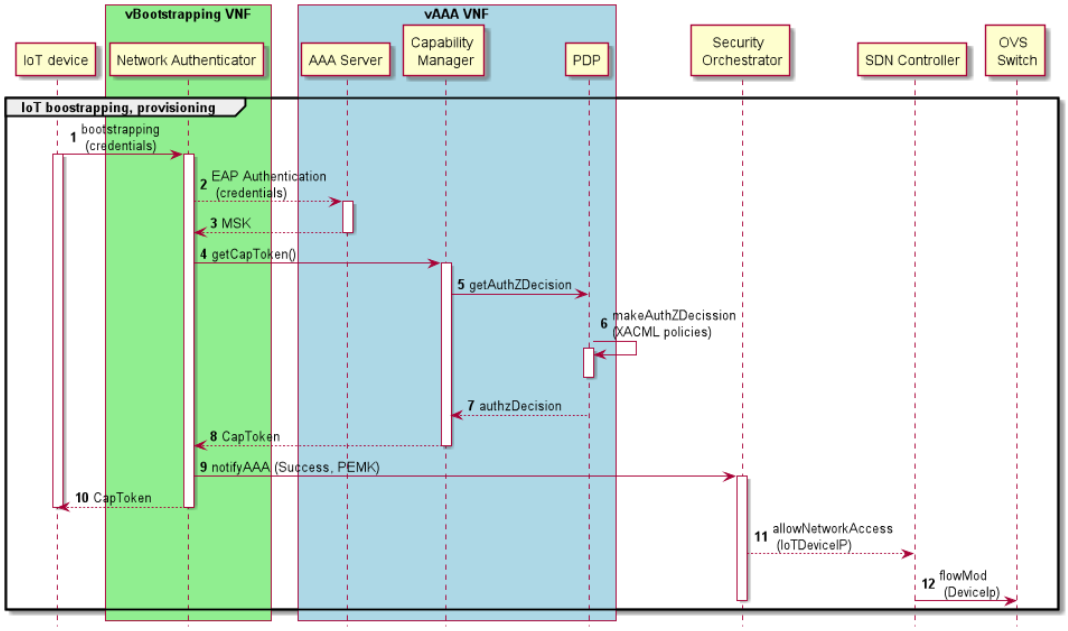

Figure 2. Bootstrapping and credential provisioning

is also deployed in the vAAA VNF, both the Network Authenticator and the IoT device share key material (MSK) (step 3) and are able to derive keys to bootstrap other Security Association Protocols (e.g. DTLS) to communicate securely with another entity (e.g. IoT broker). Additional services such as on demand credential delivery can be done using the vAAA VNF. A lightweight Capability Token (a signed assertion with the authorization claims embedded) is required by IoT devices to access to the network and publish information in the IoT broker. The token is obtained only once from the Capapbility Manager, which in turn, needs to contact the Policy Decision Point (PDP) to make the authorization decision prior delivering the token to the device. Afterwards, the SDN controller enforce network access decisions on the OVS, through Openflow, enabling the IoT device to access to the protected network.

C. Policy-based AAA management

Due to the technology's and communication's natural evolution, each time it becomes more suitable to facilitate system administrators their tasks, specially in terms of security. Security policies allow administrators abstracting themselves from the underlying systems, enabling the ability to define high-level security desires independent of the lower layers, which becomes a powerful tool for achieve interoperability and scalability. A policy framework can even provide different abstraction levels through policy refinement, facilitating the interaction with the user based on the later knowledge level. In this way, two users with different level of knowledge could model the same security policy at different levels (e.g. high and medium levels) obtaining the same effect. Of course, in the high level policy case, the system must know the missing required fields or a method to acquire them in order to be able to refine the high level policy to a medium level policy. Finally, medium level policies can be translated to a specific technical configurations over the system. The policy level separation allows abstracting policy definitions which could potentially be implemented by many different end-points leveraging on different technologies therefore making the policies entity and technology agnostic. In this sense, our solution is based on developed plugins which translate authentication, authorization, and channel protection medium-level security policies into specific device configurations (e.g. vBootstraping VNF, AAA Server, Policy Decision Point, Policy Enforcement Point and so on). The plugin selection decision is made by the Security Orchestrator who knows the current architecture status.

Following this approach, we have implemented a security framework in order to apply security policies in AAA scenarios. To this aim, our proposal extends the security policy models provided by SECURED [3] project (High-level Security Policy and Medium-level Security Policy). Specifically, we use the Medium-level Security Policy (MSPL) to model authentication and authorization policies.

The authentication security policy allows us to model information regarding authentication mechanisms to be adopted by the user (e.g. pre-shared key, certificates...), including the level of the authentication (network access, application access...). Listing 1 shows a medium-level authorization security policy example. This kind of policy

usually is compounded by a *subject*, which aims to perform some *action* over a specific target *resource*. In this case, the example is indicating *SensorA* (*subject*) is *ALLOWED* to access the resource */60001* using the *PUT* method (*action*) against the *IoT Broker* (*target*).

Listing 1. MSPL Authorization example

```
<ITResource
        ...
    <configuration xsi:type='RuleSetConfiguration'>
        <capability>
            <Name>AuthoriseAccess_resurce</Name>
        </capability>
        <configurationRule>
            <configurationRuleAction xsi:type='AuthorizationAction'>
              <AuthorizationActionType>ALLOW</AuthorizationActionType>
              <AuthorizationSubject>SensorA</AuthorizationSubject>
              <AuthorizationTarget>IoT Broker</AuthorizationTarget>
            </configurationRuleAction>
            <configurationCondition xsi:type='FilteringConfigurationCondition'>
                ...
                <packetFilterCondition>
                    <SourceAddress>SensorAIP</SourceAddress>
                    <DestinationAddress>IoT Broker IP</DestinationAddress>
                </packetFilterCondition>
                <applicationLayerCondition xsi:type="IoTApplicationLayerCondition">
                  <URL>/60001</URL>
                  <method>PUT</method>
                </applicationLayerCondition>
            </configurationCondition>
             ...
    </configuration>
</ITResource>
```

Once instantiated, the authorization medium-level security policies are translated into specific vBootstraping VNF and vAAA VNF configurations independent from the underlying technology. For instance, if the vAAA VNF instance implements an access control system based on attributes like XACML, the Security Orchestrator will choose an XACML plugin in order to translate the authorization medium-level security policy into a XACML sentences capable to configure the XACML based vAAA. In the same way, if we use PANA in order to perform the network authentication, the Security Orchestrator will choose a PANA plugin in order to perform the translation between the authentication medium-level security policy and the PANA configuration.

V. Channel protection in Softwarized IoT networks

A. Channel protection

Channel protection has become a main actor in secure communications for guarantying confidentiality and integrity. Nowadays, several techniques are available to protect the communication channel, depending on the OSI stack level we aim to protect. For instance, at network level, Internet Protocol Security (IPSec) can be applied, while at transport level, depending on the transport protocol used, Transport Layer Security (TLS) or Datagram Transport Layer Security (DTLS) could be employed. The latter guarantees equivalent security levels than TLS but using non connection oriented datagrams as underlying transport. Maintaining the security parameters, credentials and cypher-suites in large deployments can be usually considered as troublesome. Since our focus is on the IoT domain, the provision of channel protection must be designed as lighter as possible. IoT communications are in general based in datagram like communication, therefore dynamic DTLS connections have been implemented through our network infrastructure with (vBootstrapping, AAA architecture, IoT Controller and IoT Broker) dynamic key distribution.

B. Key distribution

Figure 3 shows the proposed workflow needed to perform the dynamic key distribution needed to enable the end-to-end Channel Protection. In this case, instantiating the vBootstrapping VNF in a PANA Agent. The vBootstrapping VNF notifies the result of the authentication to the Security Orchestrator, indicating the PANA Client - Enforcement Point Master Key (PEMK) derived from the MSK (Fig. 3:1). The IoT device obtains the pertinent capability tokens and it also generates the PEMK from the MSK (Fig. 3:2-3). Then, the IoT device and the Security Orchestrator generate a DTLS master key for each end-point they want to establish a DTLS connection to (Fig. 3:4-5). The Security Orchestrator notifies the generated DTLS master keys to the desired end-points through a protected channel, e.g. TLS rest API (Fig. 3:6-7). Finally, both the IoT device and the other end-point establish a DTLS connection using the master DTLS key acquired.

As a result, a softwarized, centralized and dynamic channel protection solution is obtained, leveraging on the authentication process to provide dynamic key management for M2M channel protection. In addition, the solution allows to react dynamically regenerating and redistributing a new set of keys in case of security breach.

At this point we are considering the end-points are DTLS-enabled, but this is not a mandatory condition. The end-points (including the IoT device) could be DTLS-agnostic. In this case, the framework provides a dynamic DTLS-Proxy VNF. When an end-point requires to enable a channel protection, the Security Orchestrator can request the deployment of a DTLS-Proxy as closer as possible to the end-point, and also request to the SDN controller a network configuration in order to redirect the traffic through the new VNF. Notice, that unlike in the common case without proxy, the DTLS key is delivered to Proxy-DTLS, which will establish a non-protected communication with the real end-point.

C. Channel protection policy

Similarly to the AAA case, we have followed a policy-based security management approach. The channel protection policy allows specifying different protection requirements regardless of the underlying channel protection techniques and protocols.

Listing 2. MSPL Enabling DTLS example

```
<?xml version='1.0' encoding='UTF-8' standalone='yes'?>
<ITResource
    ...
    <configuration xsi:type='RuleSetConfiguration'>
        <capability>
            <Name>Protection_confidentiality</Name>
        </capability>
        <capability>
            <Name>Protection_integrity</Name>
        </capability>
        <configurationRule>
            <configurationRuleAction xsi:type='DataProtectionAction'>
                <technology>DTLS</technology>
                <technologyActionParameters>
                    <technologyParameter xsi:type='DTLSTechnologyParameter'>
                        <localEndpoint>DTLSProxyAddress</localEndpoint>
                        <remoteEndpoint>IoTDeviceAddress</remoteEndpoint>
                    </technologyParameter>
                    ...
                </technologyActionParameters>
                <technologyActionSecurityProperty xsi:type='Confidentiality'>
                    <encryptionAlgorithm>AES</encryptionAlgorithm>
                    <keySize>128</keySize>
                    <mode>CCM</mode>
                </technologyActionSecurityProperty>
                <technologyActionSecurityProperty xsi:type='Integrity'>
                    <integrityAlgorithm>sha1</integrityAlgorithm>
                </technologyActionSecurityProperty>
            </configurationRuleAction>
            ...
        </configurationRule>
        ...
    </configuration>
</ITResource>
```

Listing 2 shows an example of channel protection medium-level security policy. This example aims to provide *Protection_confidentiality* and *Protection_integrity* between the *DTLS-Proxy* and the *IoTDevice* using *AES* as encryption algorithm with a key size of *128* bits in *CCM* mode. Once the policy has been instantiated, the Security Orchestrator decides the suitable technology to use. In case the devices involved in the DTLS connection are DTLS-enabled, the Security Orchestrator will choose a plugin in order to translate the DTLS policy to specific configuration of each involved technology, e.g. generate configurations in order to activate DTLS on an IoT device and a Context Broker Server. On the other hand, if the device is not DTLS-enabled, a DTLS-Proxy will be deployed at the edge, as close as possible to the mentioned device, and the Security Orchestrator will choose the DTLS-proxy as translator plugin. In this case, it will be also required to apply a forwarding policy in order to collocate the DTLS-Proxy in the path between the two selected devices, ideally the closer to the non DTLS-enabled device the better.

VI. Smart Building Use case

Anastacia is currently being validated in an real Smart-building scenario. The use case considers an internal attacker performing a sabotage in the building, and the ANASTACIA framework detects and reacts leveraging on the softwarized vAAA architecture proposed in this paper.

The ANASTACIA platform, trough the Monitoring service and vID, is able to detect an intrusion, and then, use the vAAA mechanism to manage dynamically the access control in the building as a reaction mechanism. To this aim, the platform firstly analyses the detected abnormalities and outliers and evaluates the severity of the situation, activating predictive mechanisms to ensure that the rest of the building operations system continues as expected.

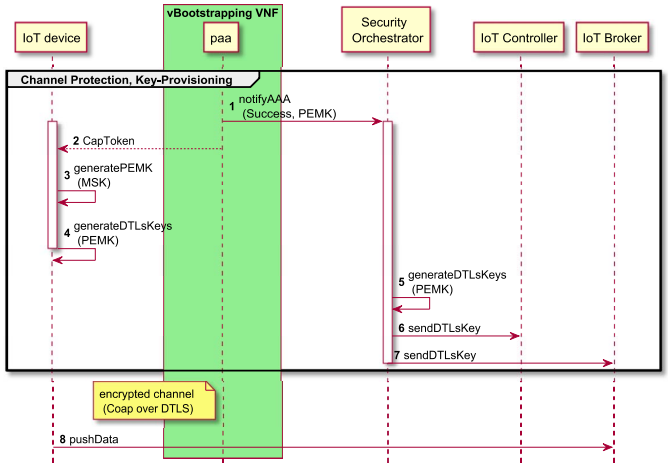

Figure 3. Channel Protection flow

The platform identities the attacks and triggers the autonomic self-healing capabilities to deploying dynamically, in the proper location, the vAAA and vBootstratpping VNF and reconfigure the system enforcing the authorization policies in the PDP, and enforcing also, through SDN, in the vSwitch the AC network rules.

Once deployed, our proposed vAAA architecture is able to detect three kind of threats regarding unauthorized attempts to access to system resources. 1) The OVS can alert when a new device tries to sends traffic to the network without previous authentication.2) The IoT Broker can detect when a new device publishes data without the required credentials for the authorization.3) An IoT device can detect when a client computer (e.g. malware) requests an actuation without the required credential for its authorization (e.g. trigger the fire alarm system).

In each detection, an alarm notification must be sent to the ANASTACIA broker indicating the type of attack and the IP source address, moreover the IP address of the IoT destination of the attack.

VII. CONCLUSIONS

This paper has introduced novel on-demand virtualized AAA and an associated channel protection mechanism specially designed to work on IoT deployments and orchestrated by a wider security architecture, the ANASTACIA framework.

Besides, the paper has described how the vAAA can bootstrap the IoT device and distribute the encryption material through the network and how DTLs channel protection is achieved. In addition, sample XACML policies describing the aforementioned mechanisms have been described while reflections on how those policies would be translated, into final technical actions on the deployment, let them be software or hardware based.

The implementation of the described systems is being carried out at the time of this writing and a pilot Testbed with real IoT devices, commodity servers to held the NFV infrastructure as well as SDN network elements is being deployed to evaluate the solution as a whole.

ACKNOWLEDGMENT

This work is the result of the stay (20177/EE/17) funded by "Fundacion Seneca-Agencia de Ciencia y Tecnologa de la Region de Murcia", under the program "Jimenez de la Espada de Movilidad Investigadora, Cooperacion e Internacionalizacion". The research has been also supported by a postdoctoral INCIBE grant within the "Ayudas para la Excelencia de los Equipos de Investigacin Avanzada en Ciberseguridad" Program, with code INCIBEI-2015-27363, as well as by the H2020 EU project ANASTACIA project, Grant Agreement N 731558.

REFERENCES

[1] J. B. Bernabé, J. M. M. Pérez, J. M. A. Calero, J. D. J. Re, F. J. Clemente, G. M. Pérez, and A. F. Skarmeta, "Security policy specification," in *Network and Traffic Engineering in Emerging Distributed Computing Applications*. IGI Global, 2013, pp. 66–93.

[2] "Deserec project: Dependability and security by enhanced reconfigurability," http://www.deserec.eu/.

[3] C. Basile, "Policy transformation and optimization techniques." [Online]. Available: https://www.secured-fp7.eu/files/secured_d42_policy_refinement_v0103.pdf

[4] "Zigbee alliance., "zigbee ip specification", zigbee document 095023r34, march 2014."

[5] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)," Internet Engineering Task Force, Internet-Draft draft-ietf-core-object-security-08, Jan. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-core-object-security-08

[6] G. Selander, J. Mattsson, and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)," Internet Engineering Task Force, Internet-Draft draft-selander-ace-cose-ecdhe-07, Jul. 2017, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-selander-ace-cose-ecdhe-07

[7] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "Sdn security: A survey," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–7.

[8] R. Lopez and G. Lopez-Millan, "Software-Defined Networking (SDN)-based IPsec Flow Protection," Internet Engineering Task Force, Internet-Draft draft-ietf-i2nsf-sdn-ipsec-flow-protection-00, Oct. 2017, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-sdn-ipsec-flow-protection-00

[9] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[10] C. Basile, A. Lioy, C. Pitscheider, F. Valenza, and M. Vallini, "A novel approach for integrating security policy enforcement with dynamic network virtualization," in *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*. IEEE, 2015, pp. 1–5.

[11] S. Ziegler, A. Skarmeta, J. Bernal, E. Kim, and S. Bianchi, "Anastacia: Advanced networked agents for security and trust assessment in cps iot architectures," in *2017 Global Internet of Things Summit (GIoTS)*, June 2017, pp. 1–6.

[12] I. Farris, J. Bernabe, N. Toumi, D. Garcia-Carrillo, T. Taleb, A. Skarmeta, and B. Sahlin., "Towards Provisioning of SDN/NFV-based Security Enablers for Integrated Protection of IoT Systems," in *IEEE Conference on Standards for Communications and Networking (CSCN-2017)*, 2017.

[13] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Internet Engineering Task Force, Jun. 2000 - ISSN: 2070-1721, updated by RFCs 2868, 3575, 5080, 6929. [Online]. Available: http://www.ietf.org/rfc/rfc2865.txt

[14] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, "Diameter Base Protocol," RFC 6733 (Proposed Standard), Internet Engineering Task Force, Oct. 2012 - ISSN: 2070-1721, updated by RFC 7075. [Online]. Available: http://www.ietf.org/rfc/rfc6733.txt

[15] B. Aboba, D. Simon, and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework," RFC 5247 (Proposed Standard), Internet Engineering Task Force, Aug. 2008 - ISSN: 2070-1721. [Online]. Available: http://www.ietf.org/rfc/rfc5247.txt

[16] D. Forsberg, Y. Ohba, B. Patil, H. Tschofenig, and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)," RFC 5191 (Proposed Standard), Internet Engineering Task Force, May 2008 - ISSN: 2070-1721, updated by RFC 5872. [Online]. Available: http://www.ietf.org/rfc/rfc5191.txt

[17] D. Garcia-Carrillo and R. Marin-Lopez, "Lightweight coap-based bootstrapping service for the internet of things," *Sensors*, vol. 16, no. 3, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/3/358