Multi-Domain Access Rights Composition in Federated IoT Platforms

Savio Sciancalepore, Giuseppe Piro, Pietro Tedeschi, Gennaro Boggia
Dep. of Electrical and Information Engineering (DEI)
Politecnico di Bari, Bari, Italy
CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni
name.surname@poliba.it

Giuseppe Bianchi
Dep. of Electronic Engineering
University of Rome "Tor Vergata", Rome, Italy
CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni
giuseppe.bianchi@uniroma2.it

Abstract

Current activities in the Internet of Things research area are devoting many efforts to the definition of architectures and mechanisms supporting the federation of heterogeneous platforms. In this context, the Multi-Domain Access Rights Composition is emerging as a promising paradigm, enabling the sharing of resources across organizations and boundaries. From the security perspective, the protection of resources against unauthorized accesses becomes even more difficult to accomplish. The work presented herein aims at solving the access control issue through a novel solution based on the Attribute Based Access Control logic. Specifically, the conceived approach leverages the Decentralized Multi-Authority - Ciphertext-Policy - Attribute Based Encryption algorithm, in a way that is completely different with respect to its conventional usage, allowing to enrich its inherent features with protection against collusion attacks, attributes revocation and preservation of user privacy. The resulting protocol offers, at the same time, the following requirements: peer authentication, data confidentiality between communicating peers, advanced access control mechanism based cryptographic algorithms, user privacy, adoption of attributes with limited lifetime, revocation of attributes, and resilience against collusion attack.

Categories and Subject Descriptors

C.2 [**Computer Communication Networks**]: Distributed Systems; K.6 [**Management of Computing and Information Systems**]: Security and Protection

Keywords

IoT, ABAC, DMA CP-ABE

International Conference on Embedded Wireless Systems and Networks (EWSN) 2018
14–16 February, Madrid, Spain

ISBN: 978-0-9949886-2-1

1 Introduction

Today, the majority of Internet of Things (IoT) platforms constitute standalone ecosystems, with proprietary methods, protocols, and interfaces [12]. In this case, a registered user is in possession of properties (i.e., credentials, permissions, or any other kind of information) that remain valid within the boundary of the IoT platform itself. Unfortunately, through these properties, the user may access to resources exposed by that IoT platform, only. Emerging services, instead, are evolving towards distributed and federated scenarios, where users may acquire properties from different domains and combine them to access to resources exposed elsewhere. To reach this goal, platform interoperability, initially discussed in [1][3][5][13][17], represents one of the most important challenging goal for many recent European initiatives. symbIoTe, BIG-IoT, INTER-IoT, FESTIVAL, VICINITY, Fiesta-IoT, to name a few.

In line with this premises, this paper focuses on an emerging paradigm, namely Multi Domain Access Rights Composition (MDARC). Initially presented in [10], it assumes that a user registered in more than one IoT platform is able to request (and obtain) the access to resources exposed by foreign IoT platforms, where it is not registered to. From the security perspective, MDARC opens the doors to important challenges related to user authentication, user privacy, collusion attacks, access control, and fine-grained and time-limited authorization.

The current literature offers many contributions that partially address some of these issues. For example, with reference to authentication and authorization services, the OAuth 2.0 authorization framework is widely used, thanks to its inherent decoupling between authentication and authorization functionalities and the possibility to be integrated with any access control logic [4][9][11]. Unfortunately, OAuth 2.0 assumes a single owner at the top of the system. Thus, it is not suited for the distributed scenario tackled in this contribution. Other contributions outsource security functionalities to the cloud. For example, this is the case of [14], [16], and [18], proposing also the adoption of the Ciphertext-Policy -Attribute Based Encryption (CP-ABE) encryption mechanism to protect resources and allow the access only to legitimate users. On the same way, [2] enriches the CP-ABE

scheme with privacy-preserving techniques and [15] periodically renews the key pairs to provide the expiration of access rights.However, being rooted on a single-authority CP-ABE scheme, these valuable contributions cannot be easily reused when considering a system with different unrelated authorities.

This paper complements the aforementioned state of the art by conceiving a novel methodology that concretely enable the MDARC paradigm in a distributed and federated IoT ecosystem. Specifically, it proposes a novel access control mechanism, based on the Attribute Based Access Control (ABAC) logic [6] and realized through the Decentralized Multi-Authority - Ciphertext-Policy - Attribute Based Encryption (DMA-CP-ABE) algorithm [7]. Differently from the current literature, the DMA-CP-ABE algorithm is used for access control purposes, rather than encryption. In addition, the resulting protocol targets, at the same time, the following requirements: peer authentication, data confidentiality between communicating peers, advanced access control mechanism based on Attribute Based Access Control (ABAC) and cryptographic algorithms, user privacy, adoption of attributes with limited lifetime, revocation of attributes, and resilience against collusion attacks.

To conclude, the present contribution is organized as follows: Section 2 describes the reference scenario and the targeted security requirements; Section 3 presents the proposed solution; Section 4 provides some interesting guidelines useful to integrate the proposed approach within the security framework designed in H2020 symbIoTe project; Section 5 tightens conclusions and draws future works.

2 Reference Scenario and Targeted Requirements

As anticipated within the Introduction, this paper focuses on the Multi-Domain Access Right Composition (MDARC) [10] paradigm and formulates a security protocol enabling a flexible access control on heterogeneous and distributed IoT resources. Specifically, MDARC leverages the Attribute-Based Access Control (ABAC) logic, and extends its functionalities for properly handling distributed and potentially large-scale scenarios .

The ABAC logic assumes to protect IoT resources through dedicated access control policies, defined as a combination of properties/access grants. To access a specific resource, in fact, a user must prove the possession of a subset of attributes that satisfies the access control policy uniquely coupled with the resource [6]. With respect to other approaches, such as Identity Based Access Control (IBAC) or Role Based Access Control (RBAC), ABAC provides better flexibility and scalability, allowing to create complex policies, neglecting details such as the number of involved users. Thus, the user is able to access a given resource only if it demonstrates to be in possession of a set of attributes that satisfy the access control policy uniquely associated with the resource; otherwise, the access is denied [6][19].

To extend the conventional ABAC logic, MDARC assumes that:

- an attribute represents a property of the user, released by a trusted entity of an IoT domain, belonging to the distributed system and simply referred to as *authority*,
- an user could be registered in one or more IoT platforms,
- an access policy can be defined by jointly considering attributes released elsewhere, and
- the user can combine attributes released by different IoT domains and request the access to an IoT resource exposed in another IoT domain.

High-level functionalities of MDARC are depicted in Figure 1. The illustrated story is very simple. A user is registered in two IoT domains, namely *platform_1* and *platform_2*. Moreover, it would like to access to resources exposed by a foreign IoT domain, namely *platform_3*. Indeed, the user performs the authentication procedure with reference attribute authorities available in both *platform_1* and *platform_2*, by using its local credentials (see step 1 and step 3 in Figure 1). If the authentication process is successfully completed, an attribute authority gives to the user a set of attributes (see step 2 and step 4 in Figure 1). Then, the user contacts the resource server of *platform_3*, which acts as a proxy and exposes the physical IoT resources of its interest. The client application delivers it the list of attributes previously received (see step 5 in Figure 1); in the case the set of provided attributes matches the access control policy associated to the requested resource, the access is granted; otherwise, it is denied (see step 6 in Figure 1).

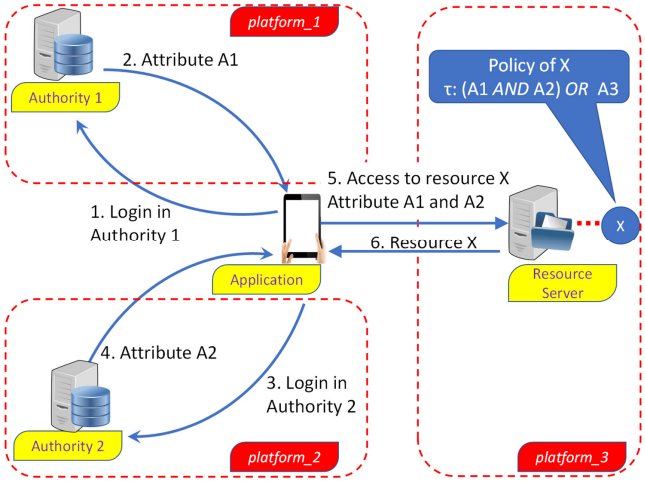

Figure 1. Reference scenario.

It is important to note, however, that the simplified approach described in Figure 1 must be revised in order to target the following security requirements:

- **Peer authentication**: each entity in the architecture must be authentic and uniquely identifiable.
- **Data confidentiality between communicating peers**: each message exchanged between authentic peers must be encrypted in order to avoid third-party eavesdroppers on the communication channel.
- **Access control based on cryptographic algorithms**: the decision on granting or denying the access must be

carried out by using cryptographic algorithms, able to guarantee the possession of necessary attributes.

- **User privacy**: the user should expose only the set attributes necessary to access the requested resources; at the same time, any tracking of its activities throughout the system must be avoided.
- **Attributes with limited lifetime**: attributes must have a well-defined time validity, after which they must be renewed.
- **Revocation of attributes**: it could be possible to revoke, at any time, the association between an attribute and the entity for which it has been released.
- **Resilience against collusion attack**: situations in which different users, registered in different IoT platforms, combine their attributes to access resources must be avoided.

3 The Proposed Approach

3.1 Network Architecture

The network architecture envisaged in this contribution integrates the following four components:

- *Client application*: it is the user that wants to access to remote IoT resources. In line with the MDARC paradigm, it can be registered in one or more IoT platforms. In every platform, the user is in possession of a set of properties (i.e., attributes), that can be used elsewhere.
- *Resource server*: it is a dedicated server exposing resources and services belonging to a given IoT platform. It processes requests coming from client applications, verifies their authenticity and provides access to resources only if the attributes presented by the client application satisfy the access control policy associated with the resource.
- *Attribute authority*: it is the trusted entity located within each IoT platform, that authenticates users and issues attributes for registered client applications. Note that attribute authorities can be unaware each of the other.
- *Identity authority*: it is a central entity, not strictly related with any of the member IoT platforms, responsible for issuing ephemeral, but unique, identities to client applications.

These components interacts each other for implementing authentication and authorization functionalities.

3.2 Standardized Approach for Peer Authentication and Data Confidentiality

Fist of all, it is important to remark that peer authentication and data confidentiality between communicating entities are not the novel security services in the envisaged architecture. Even if they are important, their fulfillment is not achieved by means of novel and innovative solutions.

From one side, in fact, it is assumed that all the aforementioned entities are equipped with a private-public key pair. For instance, U_{APP} and R_{APP} are the key pair of the client application, U_{IA} and R_{IA} are the key pair for the Identity authority, U_{AA} and R_{AA} are the key pair for the attribute authority and, finally, U_{RS} and R_{RS} represent the key pair for the resource server. The public-key is stored within a trusted X.509 certificate. Indeed, the peer authentication requirement is simply satisfied through well-known and widely accepted mechanisms.

From another side, it is also assumed that communicating peers establish a secure connection at the transport layer. Therefore, by adopting the Transport Layer Security (TLS) protocol, also the requirement related to the data confidentiality between communicating peers is reached through standardized mechanisms.

3.3 A Novel Methodology for the Access Control Procedure

The access control procedure is the novel and innovative aspect discussed in this contribution. Differently from the baseline scenario depicted in Figure 1, the access control leverages an emerging cryptographic algorithm, namely Decentralized Multi-Authority - Ciphertext-Policy - Attribute Based Encryption (DMA-CP-ABE).

DMA-CP-ABE was natively designed as an encryption scheme, able to protect a given content through an access control policy defined over a set of attributes released by trusted entities in a fully decentralized fashion [7, 8]. This contribution, instead, intentionally modifies the conventional way DMA-CP-ABE is used: the cryptographic algorithm is integrated within the access control procedure in order to allow the user to demonstrate the ownership of a set of attributes matching the access policy, without explicitly delivering the whole set of attributes in its possession to the resource server. Accordingly, also the user privacy requirement is indirectly addressed.

Figure 2 provides a preliminary overview of the designed approach. It basically extends the approach introduced in Figure 1, by means of initial security functionalities. Nevertheless, it should not be considered as a final approach presented herein. Instead, it is a simplified representation of the overall protocol discussed later on.

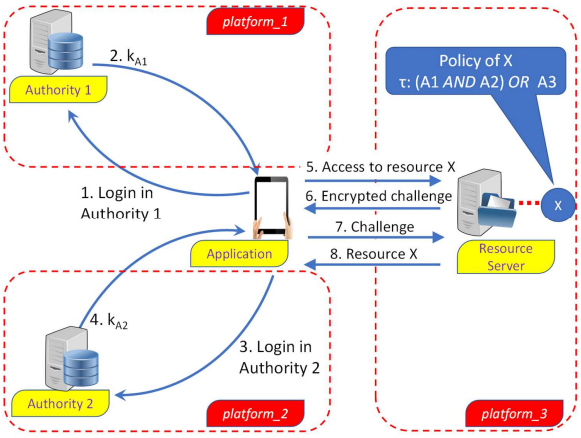

Figure 2. Reference scenario with preliminary security considerations.

According to [7] and [8], each single attribute is mapped to a set of cryptography materials, that include: public key, private key, and secret key. Let $a_{i,j}$ be the $i-th$ attribute

released by the $j-th$ attribute authority. A unique private-public key pair is assigned to this attribute, computed using arithmetic operations in a composite order bilinear group [7]. To this end, entities in the system agree on some public parameters, such as the composite prime number, the order of the bilinear groups, the generator for the group and an hash function (see [7] for more details). In particular, $R_{a_{i,j}}$ and $U_{a_{i,j}}$ are the private and public keys associated to $a_{i,j}$, respectively. While the private key is securely stored, the public key is delivered to all resource servers, thus allowing them to build access control policies that include that attribute. At the same time, DMA-CP-ABE also defines a secret key for the attribute $a_{i,j}$, namely $k_{a_{i,j}}$. It is delivered to the registered user after a successful authentication. Indeed, $k_{a_{i,j}}$ is adopted by the user for executing decryption procedures over contents protected through a given access policy.

In the conceived access control mechanism, the DMA-CP-ABE is implemented within a challenge-response scheme. Once the access policy is defined, the resource server uses DMA-CP-ABE to encrypt a random seed δ. Then, the resource server sends the random seed to the client application. Now, let τ and $\mathcal{U} = \{U_{a_{i,j}}\}$ be the access policy and the set of public keys associated to the attributes considered by the policy. The encryption procedure of the random seed δ produces the challenge σ, as reported below:

$$\sigma = E_{DMA-CP-ABE}[\delta, \tau, \mathcal{U} = \{U_{a_{i,j}}\}]. \quad (1)$$

The client application adopts the DMA-CP-ABE algorithm and the set of secret keys in its possession, i.e., $\mathcal{K} = \{k_{a_{i,j}}\}$, to decrypt the received challenge. In this case, the decryption procedure could be formally described as in the following:

$$\delta = D_{DMA-CP-ABE}[\sigma, \tau, \mathcal{K} = \{k_{a_{i,j}}\}]. \quad (2)$$

The access to the resource is granted in the case the decryption process ends successfully and the client application is able to send back to the resource server the original random seed. In fact, a successful decryption process demonstrates the possession of the set of cryptographic materials related to a subset of attributes matching the access policy.

All the details related to cryptography operations executed in both Eq. (1) and Eq. (2) are discussed in [7] and [8].

However, it is important to remark that, despite its inherent protection against the key escrow problem, the DMA-CP-ABE algorithm cannot be used as it is. In fact, DMA-CP-ABE does not natively offer the protection against collusion attacks. At the same time, it does not support the usage of attributes with limited lifetime. Thus, the high-level solution depicted in Figure 2 and summarized before is properly extended to embrace further functionalities. The possibility to manage attribute revocation and to masquerade the identity of the client application is also taken into account. The resulting protocol is described in the next sub-section.

3.4 Detailed Description of the Protocol

As illustrated in Figure 3, the designed protocol is implemented in four different phases, that are (i) setup, (ii) ephemeral identity generation, (iii) authentication phase, and (iv) authorization phase. Each step include many atomic operations, as discussed in the sequel:

- **Setup phase.** In this phase, attribute authorities generate attributes and their related cryptographic material; at the same time, resource servers configure access policies for the IoT resources they expose.
- **Ephemeral identity generation phase.** The client application initially contacts the identity authority to retrieve an ephemeral identity, uniquely associated to its real identifier I_{APP}. The ephemeral identity is adopted within the DMA-CP-ABE algorithm to offer the protection against collusion attacks and the usage of attributes with limited lifetime. The client application initially sends its X.509 certificate to the identity authority. The identity authority verifies the authenticity of the certificate and randomly extracts an ephemeral identity ε. Then, it generates an ephemeral attribute, a_ε, with its related private and public keys set to R_{a_ε} and U_{a_ε}, respectively, as well as the secret key set to k_{a_ε}. The identity authority uniquely binds the real identity of the client application with the ephemeral identity ε through an hash function: $H(I_{APP}||\varepsilon)$. Then, it calculates a proof message containing the output of the aforementioned hash function, the ephemeral identity ε, the secret key associated to the ephemeral attribute k_{a_ε}, the public key of the ephemeral attribute U_{a_ε}, and the time validity of the ephemeral identity T. The whole proof is encrypted through the private key of the identity authority, i.e., R_{IA}:

  $$proof = E[\{H(I_{APP}||\varepsilon), \varepsilon, k_{a_\varepsilon}, U_{a_\varepsilon}, T\}, R_{IA}] \quad (3)$$

  Finally, the proof is delivered to the client application, that can easily verify its integrity and authenticity. Note that the proof does not explicitly contain the identifier of the client application, given that it is hashed with the ephemeral identity. Thus, user privacy is achieved thanks to the fact that it is not possible to use the proof to track user's activities within the system.
- **Authentication phase.** The client application logins in each of the attribute authorities where it is registered to, in order to obtain its attributes.

  First, the client application provides to the $j-th$ attribute authority its own credentials (e.g., username and password), the proof received by the identity authority, and the X.509 certificate of the identity authority. The $j-th$ attribute authority verifies the integrity and authenticity of the proof, and checks that it has not expired (thanks to the time validity field). If there are no errors, the attribute authority delivers the attributes to the client application. According to the DMA-CP-ABE technique, the attribute $i-th$ $a_{i,j}$ is released by the $j-th$ attribute authority as a secret key, $k_{a_{i,j}}$, generated from the private key of the attribute authority and the ephemeral identity ε (details in [7], [8]). Finally, attributes are delivered to the client application.At the end of the authentication phase, the client application owns a wallet of attributes, and its related cryptography

material.

- **Authorization phase.** The client application sends a request for a given resource to the resource server, alongside the proof message generated by the identity authority and the X.509 certificate of the identity authority. The resource server verifies the validity of the proof and extracts the ephemeral identity ε. Moreover, it creates a new ephemeral policy τ_ε, by combining the policy originally assigned to the resource, i.e., τ, and the ephemeral attribute ε. This is performed in order to verify that (1) the client application possesses the attributes required to access the resource and (2) these attributes are fresh, in the sense that they are associated to a valid ephemeral identity, not yet expired. The resulting policy is:

$$\tau_\varepsilon = \tau \,\text{AND}\, \varepsilon. \tag{4}$$

In line with the high-level approach described in the previous sub-section, the resource server extracts a random number δ and generates a challenge σ, i.e.,

$$\sigma = E_{DMA-CP-ABE}[\delta, \tau_\varepsilon, \mathcal{U} = \{U_{a_{i,j}} || U_{a_\varepsilon}\}]. \tag{5}$$

Then, it sends the challenge σ and policy τ_ε to the client.

The client application decrypts the received challenge, as in the following:

$$\delta = D_{DMA-CP-ABE}[\sigma, \tau_\varepsilon, \| = \{k_{a_{i,j}} || k_{a_\varepsilon}\}]. \tag{6}$$

Indeed, the client application delivers the computed value δ to the resource server. The access to the target resource is authorized in the case the received value is equal to the one previously extracted by the resource server. Otherwise, the access is denied.

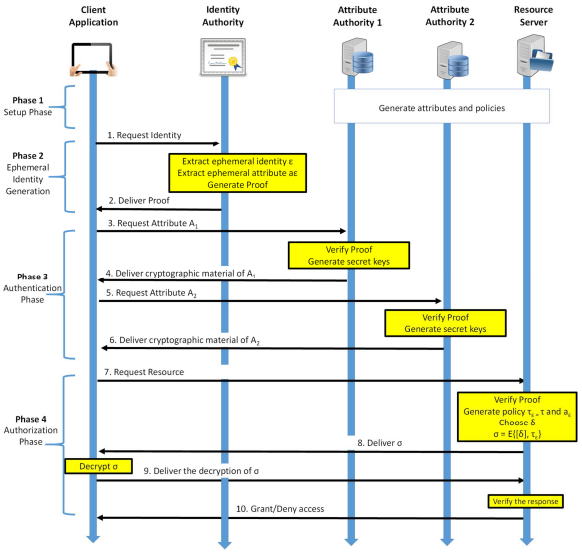

Figure 3. Detailed description of the protocol.

3.5 Further Considerations on Attribute Revocation and Offline Scenario

Differently from other approaches available in the literature and based on the DMA-CP-ABE algorithm, the conceived protocol also supports attributes revocation. In fact, the identity authority can handle an Identity Revocation List, containing all the identities whose validity has been revoked before the legacy expiration date, indicated within the proof. The Identity Revocation List is periodically updated by the identity authority, based on anomalies detected throughout the system. Each resource server can periodically download the revocation list, and block all current sessions and future access to resources when unauthorized requests occur.

Also, the system easily supports offline access to resources. Supposing that the ephemeral identity has a suitable time duration, the client application can collect the attributes when it is online and access resources hosted on a resource server not connected to the Internet. Obviously, this scenario hinders the possibility to block identities that have been revoked, because in this case the most updated Identity Revocation List cannot be downloaded. However, the resource server can report the anomaly when it comes back online.

4 Integration in H2020 SymbIoTe

H2020 symbIoTe is an European project focusing on the design of a mediation framework enabling the collaboration of vertical IoT platforms. It proposes a flexible security framework supporting authentication and authorization features in a distributed environment, like the one taken into account in this contribution. In particular, the security framework integrates the following components and libraries: (1) Core Authentication and Authorization Manager (CAAM), that provides platforms authentication and users registration in symbIoTe core services, guest user management, administration management of platforms, management of local user, attributes, home/foreign tokens, and X.509 certificates; (2) Platform Authentication and Authorization Manager (PAAM), which provides the same functionalities of CAAM, but within a given platform; (3) Security Library, offering cryptography operation (as challenge-response procedure, check access policy procedure) and security related API for symbIoTe components; and (4) Anomaly Detection module, able to detect malicious and unknown threats.

Assuming that an user is registered in *platform_1* and would like to access to resources in *platform_2*, the procedure envisaged by the H2020 symbIoTe project is described in the sequel. The user contacts the PAAM of *platform_1* and performs the authentication process. The PAAM of *platform_1* releases a home token with a set of attributes associated to the user. Then, the user contacts the PAAM of *platform_2* and performs a foreign authentication process. Indeed, the PAAM of *platform_2* releases a foreign token, storing a set of attributes assigned to the user, but usable within *platform_2*. Finally, the user contacts the Resource Access Proxy (RAP) of *platform_2*, delivers to it the aforementioned foreign token and details the access request. The user demonstrates to be the real owner of the token through a challenge-response scheme. The RAP allows the access to the resource only if the set of attributes stored within the foreign token matches the access policy protecting the requested resource. It is important to note that this procedure guarantees all the requirements listed in Section 2, excepting fro those related to the user privacy and the provisioning of access control procedure through cryptographic techniques.

All the attributes, in fact, are stored in clear within both home and foreign tokens and delivered to the RAP every time the user issues an access request. Nevertheless, the proposed approach can be integrated within the security framework of H2020 symbIoTe project, thus addressing also these important requirements. To this end, it is necessary to map components and functionalities, as in the following: (i) the proposed client application represents the symbIoTe application, (ii) the proposed resource server represents the symbIoTe RAP; (iii) the functionalities of the proposed attribute authority can be integrated in the symbIoTe Platform-AAM; and (iv) the functionalities of the proposed identity authority are integrated in the symbIoTe Core-AAM. In addition, it is necessary to introduce few enhancements to the resource access procedure described before, as summarized below. The user contacts the CAAM and performs the authentication process by requesting an ephemeral identity. The generated proof is inserted in a JSON Web Token (JWT) token namely proof token, signed by the CAAMs private key, and sent to the user. The user contacts the PAAM of *platform_1* and performs the authentication process, based on the the proof token previously received. The PAAM of *platform_1* releases a set of attributes associated to the user in the form of secret keys, calculated by jointly considering the attributes themselves and the ephemeral identity. These attributes are still stored within the home token. Then the user contacts the PAAM of *platform_2* and performs the foreign authentication process, by sending proof token and the home token. The PAAM of *platform_2* creates a foreign token with attributes available in the home token. The user contacts the RAP of *platform_2*, and delivers to it the aforementioned proof token and the resource access request. The RAP initiates the challenge-response scheme presented in this contribution in order to verifies that the user owns the right secret keys. Note that this challenge-response is completely different with respect to the one already conceived in H2020 symbIoTe project. The RAP allows the access to the resource only if the challenge is properly decrypted by the user, matching the access policy that protects the requested resource. As a result, this solution permits to reduce the data transmission overhead, by ensuring an efficient, secure, and privacy oriented access control mechanism.

5 Conclusions and Future Works

This work presented a novel access control mechanism for federated IoT platforms, based on the ABAC logic and the DMA-CP-ABE algorithm. The resulting solution is able to guarantee, at the same time, peer authentication, data confidentiality between communicating peers, advanced access control mechanism based cryptographic algorithms, user privacy, adoption of attributes with limited lifetime, revocation of attributes, and resilience against collusion attacks. Future research activities include the implementation of the conceived approach and the evaluation of its performance through experimental testbeds consisting of real interoperating IoT platforms.

6 Acknowledgments

This work was framed in the context of the project SymbIoTe, which receives funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 688156.

7 References

[1] A. Bröring, S. Schmid, and C. K. S. et al. Enabling IoT Ecosystems through Platform Interoperability. *IEEE Software*, 34(1):54–61, Jan. 2017.

[2] C. Buttner and S. A. Huss. Attribute-based authorization tickets for Car-to-X communication. In *IEEE Conf. on Commun. and Network Security (CNS)*, pages 234–242, Oct. 2016.

[3] A. Celesti, M. Fazio, and M. G. et al. Characterizing Cloud Federation in IoT. In *Int. Conf. on Adv. Inform. Netw. and Appl. Workshops (WAINA)*, pages 93–98, Mar. 2016.

[4] S. Cirani, M. Picone, and P. G. et al. IoT-OAS: An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios. *IEEE Sensors J.*, 15(2):1224–1234, Feb. 2015.

[5] M. Ganzha, M. Paprzycki, W. Pawowski, P. Szmeja, and K. Wasielewska. Semantic interoperability in the Internet of Things: An overview from the INTER-IoT perspective . *Journal of Network and Computer Applications*, 81:111 – 124, 2017.

[6] V. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Nist special publication 800-162, NIST, Jan. 2014.

[7] A. Lewko and B. Waters. Decentralizing Attribute-based Encryption. In *Proc. of the 30th Annual Int. Conf. on Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, pages 568–588, 2011.

[8] A. Lewko and B. Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *in Proc. of CRYPTO*, pages 180–198, 2012.

[9] C. Pisa, A. Caponi, T. Dargahi, G. Bianchi, and N. Blefari-Melazzi. WI-FAB: Attribute-based WLAN Access Control, Without Pre-shared Keys and Backend Infrastructures. In *Proc. of the ACM Int. Worksh. on Hot Topics in Planet-scale mObile Computing and Online Social neTworking*, pages 31–36, 2016.

[10] S. Sciancalepore, M. Pilc, S. Schroder, G. Bianchi, G. Boggia, M. Pawlowski, G. Piro, M. Plociennik, and H. Weisgrab. Attribute-based access control scheme in federated iot platforms. In *Proc. of 2nd Workshop on Interoperability and Open-Source Solutions for the Internet of Things*, Stuttgart, Germany, Nov. 7 2016. Springer.

[11] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi. OAuth-IoT: An access control framework for the Internet of Things based on open standards. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 676–681, Jul. 2017.

[12] K. J. Singh and D. S. Kapoor. Create Your Own Internet of Things: A survey of IoT platforms. *IEEE Consumer Electronics Magazine*, 6(2):57–68, Apr. 2017.

[13] S. Soursos, I. P. Zarko, and P. Z. et al. Towards the cross-domain interoperability of IoT platforms. In *European Conference on Networks and Communications (EuCNC)*, pages 398–402, 2016.

[14] L. Wang, Q. Xie, and H. Zhong. Cooperative Query Answer Authentication Scheme Over Anonymous Sensing Data. *IEEE Access*, 5:3216–3227, Mar. 2017.

[15] K. Yang, Z. Liu, X. Jia, and X. S. Shen. Time-Domain Attribute-Based Access Control for Cloud-Based Video Content Sharing: A Cryptographic Approach. *IEEE Trans. on Multimedia*, 18(5):940–950, May 2016.

[16] L. Y. Yeh, P. Y. Chiang, Y. L. Tsai, and J. L. Huang. Cloud-based Fine-grained Health Information Access Control Framework for Lightweight IoT Devices with Dynamic Auditing and Attribute Revocation. *IEEE Trans. on Cloud Computing*, PP(99):1–1, Oct. 2015.

[17] I. P. Zarko, S. Soursos, and I. G. et al. Towards an IoT framework for semantic and organizational interoperability. In *Global Internet of Things Summit (GIoTS)*, pages 1–6, Jun. 2017.

[18] S. Zhou, R. Du, J. Chen, J. Shen, H. Deng, and H. Zhang. Facor: Flexible access control with outsourceable revocation in mobile clouds. *China Communications*, 13(4):136–150, Apr. 2016.

[19] Y. Zhu, D. Huang, C. J. Hu, and X. Wang. From RBAC to ABAC: Constructing Flexible Data Access Control for Cloud Storage Services. *IEEE Trans. on Services Computing*, 8(4):601–616, Jul. 2015.