Securing An Agri – Food Marketplace: An Implementation of a Robust Security Layer with API Gateway Integration

Nikos Papageorgopoulos
Robotics & Cognitive Systems Unit
UBITECH
Athens, Greece
npapageorgopoulos@ubitech.eu

Danai Vergeti
Robotics & Cognitive Systems Unit
UBITECH
Athens, Greece
vergetid@ubitech.eu

Elena Politi
Robotics & Cognitive Systems Unit
UBITECH
Athens, Greece
epoliti@ubitech.eu

Dimitris Ntalaperas
Robotics & Cognitive Systems Unit
UBITECH
Athens, Greece
dntalaperas@ubitech.eu

Eleni Tsironi
Project Management Unit
& Agri Food Tech Lead
UBITECH
Athens, Greece
etsironi@ubitech.eu

Xanthi S. Papageorgiou
Robotics & Cognitive Systems Unit
UBITECH
Athens, Greece
xpapageorgiou@ubitech.eu

***Abstract*—As food safety is undergoing through significant challenges due to recent food scandals, and the consumers demands for products of higher quality is increasing, the need for better knowledge of the food production processes and adoption of data sharing practices in the product and supply chain management are emerging. To address those issues, data sharing platforms have been introduced as essential tools for creating high value from data with secure and mutually beneficial multi-partner data sharing fascilitation. Blockchain technology, through its inhereted distributed nature can help to build trust mechanisms to enhance transparency and security dimension of food chains. In this work we propose a novel security mechanism for proper authentication and authorization when accessing resources through an agrifood data platform. Our proposed methodology aims to deliver sophisticated backbone service capabilities that will enable trusted, secure, automated, robust and controlled data transactions for food certification to all food sector businesses that demand easy, fast, and actionable access to variegating food safety data from multiple devices and in various settings.**

***Index Terms*—food safety, data platform, certification, security mechanisms**

I. Introduction

The food sector is going through tremendous challenges with a series of food scandals and controversies taking place in recent years, while the fight against food waste is still one of the biggest concerns global policy makers are trying to solve. What's more, data localisation restrictions and legal uncertainties in the data economy of the food sector bring several challenges to data sharing practices in the product and supply chain management. Certain initiatives, such as the Global Food Safety Initiative (GFSI) [1] are opening the way for international alignment of food safety standards, ensuring the safe delivery of food to all consumers internationally, and providing an international stakeholder platform for collaboration, knowledge exchange and networking in the area of food safety. However, the compliance of the agri-food and grocery sector with the GFSI food safety standards is moving at a slow pace. This increases the pressure of robust food safety data exchange in a timely, trusted and secure manner for all stakeholders in the supply chain [2]. What's more, concerns over credit and recognition, misinterpretation, loss of control, lack of resources, socio-cultural factors and ethical and legal barriers may impede stakeholder's decision and the mode on data sharing broadly [3]. Blockchain technologies, which are inherently distributed by design, is a promising solution for facilitating trusting mechanisms through the full information transparency and security of data in the agri-food supply chains [4]. Since agri-food trade involves several stakeholders, the assignment of unique digital identifiers to products would make these traceable through supply chains, and in turn prevent food waste, allows consumers to work out the ecological footprint of their food and guides the distribution of food surplus [5].

In previous work authors presented a semantic data platform based on Blockchain technology for facilitating trusted, secure, automated, robust and controlled data exchange that is critical to food certification, through a shared reference architecture and a set of common governance rules [6]. In this paper, we delve into the security mechanism for transparent data transactions in the food supply chain that ensures proper authentication and authorization when accessing resources. Our methodology refers to different layers for data security and privacy assurance: (a) end-to-end hybrid encryption for data assets (before, during and after their uploading in the TheFSM platform) and secure tunnels for direct key sharing

to authorized data consumers with active data contracts, (b) attribute-based access control policies that formally describe the circumstances under which access requests to data assets should be granted, and are easily interpretable into policy enforcement rules; (c) multiple data anonymization methods and guidelines for data providers.

The rest of this work is as follows. In Section 2 that follows we summarize the main research areas related to data sharing practices in the product and supply chain management. Section 3 provides details on the methodology that we propose and specifically the authentication and authorization mechanisms. Finally, Section 4 provides a discussion on the results achieved so far and describes our next steps.

II. Related Work

Sharing information between partners across the supply chain offers great benefits achieving competitive product delivery, elevation of digital platform business models and enhancement of operational efficiencies [7]. Data sharing relies on three main pillars, namely the partners' willingness to share information, the existence/adoption of adequate information technology, and identifying the appropriate information to share [8]. In this context, data sharing platforms can create value for the participant stakeholders from collecting, integrating, and sharing different types of data. but to achieve additional benefits all parties along the supply chain will have to participate in data sharing and invest in additional data capturing.

In the food safety sector, there is a need to represent all food safety standards and their specifications for data monitoring and collection as commonly referenced and interoperable information models that can link, map, translate and transform different data formats in equivalent versions and formats [9] . Blockchain technology can also facilitate information transparency and security in food supply chains by its inherently distributed nature [4], [9]. A blockchain based platform for data management and multi-partner data sharing architecture where data subjects can control access to their data is proposed by [10]. The operations of the proposed platform are also compliant with the General Data Protection Regulation (GDPR) conditions. The data trusts concept has been explored by [7]. In this work, authors proposed a digital platform that reinforces multi-partner data sharing, where data subjects can manage access and usage of their information based on policies. Another interesting study developed a farm management system that handles digital data about the feeding and animal breed age and health through blockchain technology [11]. In spite of the extend of existing and emerging technologies, we are still far from converging on an established model of a Data Trust, due to stakehoder's lack of trust for collating, and sharing the data or due to lack of avilable data [12].

In our work we propose a security methodology that comprises two mechanisms that ensure proper authentication and authorization when accessing resources through our data platform, namely the ABE (Attribute based encryption) which addresses the issue of encrypting documents and data according to a set of attributes, and ABAC (Attribute-Based Access Controller) which addresses the authorization aspect of accessing resources, based on both environmental and user-specific attributes. Our proposed model provides a well-constructed data management roadmap that is implemented within our data sharing platform in order to ensure that all data processing adheres to the required technical safeguards, such as security, data processing, data curation, provenance, ownership etc.

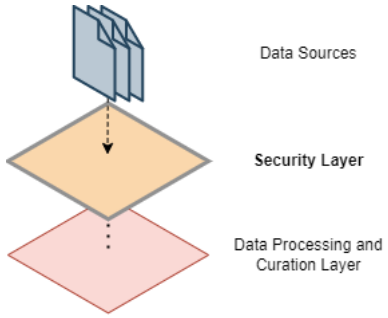

Fig. 1. Integration of security among other architectural layers

III. Methodology

This section is devoted to describing the crucial technologies that are utilized in tandem to ensure proper authentication and authorization when accessing system resources. The ABE (Attribute-Based Encryption) which addresses the problem of encrypting documents and data based on a set of attributes, and ABAC (Attribute-Based Access Controller) which addresses the authorization aspect of accessing resources based on both environmental and user-specific attributes. Figure 2 depicts the security layer which comprises of the components that are described in the following section

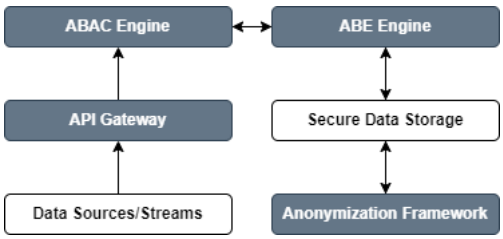

Fig. 2. Security Layer Tech Stack

A. Authentication

Technologically, there are well-known industry standards enforcing this, such as OAuth [13] and and JSON Web Tokens (JWTs) [14]. OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information

about their accounts with third party applications or websites. Generally, OAuth provides clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without providing credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server. JSON Web Token (JWT) is an Internet standard for creating data with optional signature and/or optional encryption whose payload holds JSON that asserts some number of claims. The tokens are signed either using a private secret or a public/private key. For example, a server could generate a token that has the claim "logged in as admin" and provide that to a client. The client could then use that token to prove that it is logged in as admin. The tokens can be signed by one party's private key (usually the server's) so that party can subsequently verify the token is legitimate. If the other party, by some suitable and trustworthy means, is in possession of the corresponding public key, they too are able to verify the token's legitimacy. The tokens are designed to be compact, URL-safe, and usable especially in a web-browser single-sign-on (SSO) context. JWT claims can typically be used to pass identity of authenticated users between an identity provider and a service provider, or any other type of claims as required by business processes.

B. ABAC model, Authentication Authorization Engine

To represent the ABAC access controller, we utilize the PERM meta model which is already built-in in *Casbin*, the ABAC library used for authorization in our implementation.

In this model, each request is essentially a triplet which comprised of the following entities

- Subject: This is a POJO object containing all attributes uniquely defining the user.
- Verb: It represents anything the request is acting on. For our purposes, this is a resource. Resources are represented as URLs (due to REST principles).
- Action: The desired action of the request.

Casbin's meta-model contains a set of user-defined policies, each policy including the following: A condition that must be met, combining boolean conditions of the user's attributes (not all of them are necessary to appear in the policy, but they can).

- The resource is protected by the policy.
- The action involved with this policy.
- The effect of the policy (allow or deny access)

When a request arrives, the engine will filter the request against policies that match a regular expression (simply put, the request must match relevant policies, i.e., accessing the same resource with the same action, boolean conditions being met). Then, we only keep the matching policies and evaluate the results in order to obtain their effect. Finally, the effects will indicate whether the request will be granted or not, depending on the effect expression.The effect expression is also known in other schemes as the unification algorithm of the policies. It defines whether the access request should be approved if multiple policy rules match the request. The supported effect expressions are:

- Allow-override: If even one matching policy has allow as effect, the request is granted access.
- Deny-override: If even one matching policy has deny as effect, the request is immediately denied access.
- Allow-and-deny: If a request has at least one matching policy with allow as effect and no deny effects are matched as well, the request is granted access.
- Priority: The first effect encountered (allow or deny) determines whether the request will be granted access.

In order to maximize flexibility and ensure robustness of policy evaluation, we use the allow-and-deny effect expression. Additionally, to guarantee the maximum amount of fine-grained control in the system, we use ABAC. Currently, the attributes accompanying a user are:

- Their ID (unique identifier)
- The UNIX timestamp of the request
- The user's roles
- The year of the request
- The month of the request
- The day of the request
- The day name of the request (this is useful for policies denying access in specific days by name)
- The hour of the request
- The minute of the request

C. Data Encryption

An important facility offered by the system is data encryption. Due to GDPR regulations and the fact that we deal with sensitive information, legal contracts, and protected datasets, it is paramount to ensure the data remains protected. To that end, we guarantee that while data is being transferred between the end user and the platform that has implemented this security layer, it is always encrypted. In the next parts, we describe encryption schemes, discussing our proposed protocol.

D. RSA Encryption

When encrypting data, one of the most frequently used ways is RSA encryption [15]. In RSA encryption, each member of the protocol has a pair of keys, a private and a public key. The private key must be kept private at all times; only its owner should have it. On the other hand, the public key is available to all interested parties. When needing to send a message to someone, the member asks for their public key and encrypt the message using the public key they received. When the other member receives the message, they decrypt it using their own private key. The stark difference of how both keys are utilized in the protocol is the reason this encryption is also known as Asymmetric Encryption.

E. Symmetric Encryption

Following the previous description of RSA, it is paramount to explain why it is not enough for the platform's needs. A major shortcoming of RSA is that, in order to encrypt a message, the key needs to be almost as large as the message itself, raising serious issues during data transfer and size. RSA's shortcoming can be overcome by another scheme, Symmetric Encryption. Instead of having a pair of keys per member of the protocol, all involved parties should use the same key and which they have obtained before the protocol starts. It is called symmetric encryption because the same key both encrypts and decrypts.

F. Attribute-Based Encryption

Finally, for the sake of completeness we present an even more advanced encryption scheme and justify why we opted against it. Attribute Based Encryption (ABE) is a concept introduced in 2004 by Sahai and Waters [16]. Based on IBE, it allows a user to encrypt data so that it can only be decrypted by users with certain attributes. Specialized protocols and adaptations of ABE exist, such as FAME (Fast Attribute-based Message Encryption), DAC-MACS (Data Access Control for Multi-Authority Storage Systems), RD-ABE (Revocable and Decentralized Attribute Based Encryption), CP-ABE (Ciphertext-Policy Attribute Based Encryption) and KP-ABE (Key-Policy Attribute Based Encryption). In many ways, this is similar to the ABAC protocol in request authorization. Once again, a user must have a specific set of attributes validated against a complex boolean policy, being able to decrypt the data iff they match the policy filter. While it adds even more robust data encryption on top of our proposed hybrid protocol, there are a few reasons we opted against it. The first reason lies in an inherent problem ABE protocols have, which is the immutability of the initial attributes defined by ABE. Once these attributes are set, encryption takes them into consideration and therefore any change to the attributes immediately invalidates the entire protocol. Furthermore, ABE schemes have difficulties with access revocation due to the encryption depending on a setup phase for the protocol. Once a user has the proper attributes to decrypt a resource, even if they are revoked access, they can still decrypt the resource. Additionally, due to the nature of ABE schemes, it is difficult to have multiple authorities enforcing the protocol. While some versions of ABE remedy this, it still adds multiple layers of complexity on an already complex protocol. Moreover, some attributes in policies fitting for a Data Market will naturally involve information such as dates. Keeping in mind that ABE cannot add or remove attributes once set up, it is impossible to have such policies in the encryption level.

G. API Gateway

Generally speaking, an API Gateway is an API management tool which sits between a client and a collection of backend services, which serves as a reverse-proxy accepting all API calls, aggregating services (if necessary) and returns the obtained result.

The necessity of the API Gateway for the system originated from two aspects: A way to share data assets via APIs, as well as discovering these APIs themselves was required. General services such as analytics, third parties offering their own services as part of the platform for added value can be integrated into the platform via the API gateway.

Apart from those aspects, additional reasons we opted to introduce the API Gateway are the following:

- Protection of APIs from overuse and abuse. This is ensured by authenticating and authorizing requests.
- The potential of running analytics on top of requests.
- The potential of monetized APIs and billing.
- The ability to call multiple micro-services to cover the needs of a single request. Due to having many micro-services throughout a platform, it is necessary to call many of them for a single request
- Addition, removal or update of all services is handled at this single point.

The API Gateway is a backend service which can also be handled via the platform's user interface for convenience. When a new API is added to the gateway, the API is defined as a new endpoint, requiring the user to provide the URL of the API (parametric URLs for REST APIs are fully supported as well), the method of the API (POST, GET, DELETE, etc.), a small description, the authentication method the system needs to use to call the API (e.g., the API in question could be protected by API key or JWT) and the service under which this API is provided (services are a level above endpoints to group APIs of the same provider and to make searching for APIs easier). Once the API is added, the user can then set up ABAC policies to restrict access to that API. These policies are subject to the exact same constraints as the policies enforced everywhere else throughout the platform. When data assets are offered by API instead of being static and uploaded directly to the system, they are going to be automatically added under the hood to the API gateway as services (the user will of course be notified and asked for approval of that action beforehand), enabling data exchange subject to regular ABAC policies. The robustness of the gateway is evident by the workflow which illustrates how a request is processed, when calling the API gateway

- The user interested in consuming data from a specific API calls the API gateway, asking it to call on their behalf the desired API (they are responsible for providing their own JWT, the URL of the API they want to use, its method and also any body that should be sent, for POST/PUT requests).
- The API Gateway will use the user's JWT to filter the request before submitting it, to ensure the requester is both authenticated and authorized by the platform.
- Upon successful authentication and authorization, the gateway will use all input provided by the requester and will call the API in question. If the API itself is somehow protected, it will use the declared API key or JWT as part of the request.

- Once the request returns the response to the API Gateway, it will return the response to the caller.

H. Anonymization Framework

Pseudonymisation refers to procedures where sensitive data are mapped to generic values, so that they can be protected, while anonymization maps sensitive data to generic, random values. Assuming "X" was originally ID=3, the main difference lies in the fact that with the former it is still possible to deduce that "X" refers to the same information every time "X" is encountered, while the latter can map 3 to "X", "Y", "Z" for every time it is encountered throughout the text. Pseudonymisation obviously exposes some knowledge about the original data, however this can be useful. For example, risk estimation can take into consideration sensitive data about companies which are pseudonymized and conduct a thorough analysis, without ever exposing their identities.

1) Design and Functionalities Overview: The initial approach for the security layer is the utilization of pseudonymisation, which can then be adapted into full anonymization, should the needs of the project require it.

The Anonymisation component is responsible to implement the pseudonymisation and anonymisation of the platform data. The component includes the following sub-components:

- Consent database: A database which stores the data subjects who have provided consent to the proposed platform.
- Framework database: A database which contains the PIIs (Personally Identifiable Information) of all the data subjects.
- Re-identification database: A database which contains the original data of the data subjects or other data which can be used to match the pseudonymised (or anonymised) data to the data subjects. These data need to be pseudonymised (or anonymised) and their access is restricted only to the authorised personnel.
- Exposed database: A database which contains the pseudonymised data which are accessed and disseminated to the various parties which may use the system.
- Pseudonymisation: A component which will perform pseudonymisation transformations on the data.
- Anonymisation: A component which will anonymise the data.
- Data adapter: A software component which is responsible to implement the pseudonymisation of the data.

The pseudonymisation process is briefly described below:

When collecting personal data, the Data Adapter will query the Consent database and the Framework database. The consent database will have a map of all subjects that have provided consent to our implementation. The Framework database will contain the PIIs of all data subjects. If confirmation from the consent or the framework database occurs, the Pseudonymisation Module will perform pseudonymisation on the data; it will store the pseudonymised data in an open dataset that can generally be accessed by parties being in communication with the platform; and it will store the re-identification data in a separate database; the Re-identification database. The Re-identification database will not be publicly accessed but will be used and maintained by each of the data controller's users. When re-identification is needed at run-time (e.g., when the email of a user needs to be verified), the Pseudonymisation Module will communicate with the Re-identification database to obtain the original data; apart from this case, access to the re-identification database will be restricted.

After storage, an extra Anonymisation module will provide the functionality of generating anonymised data from the exposed data set. The implementation of the anonymisation module will be based on the ARX Framework and will produce a data set with high k-value, l-diversity and t-closeness parameters. In case the platform operator imports a population table, the Anonymisation module will also produce a low value of (for the specifics of k-value, l-diversity, t-closeness and - difference. The anonymised data set will contain all useful information regarding user actions and cases and can still be used to compute analytics and provide useful feedback. Since data subjects cannot be de-identified from the anonymised data set, it can be stored or archived regardless of the status of consent forms.

In case that a subject is removed from the framework database or a consent is revoked, the Pseudonymisation Module will remove for this subject the re-identification data form the re-identification database. The pseudonymised data will be automatically converted to anonymous data upon this removal, so they can still be stored in the Exposed database. Upon revocation of consent, the deletion of re-identification data may take some time due to the system having to poll the consent database and the technical expert receiving the notification to delete re-identification data. This will be explicitly noted in the consent form. The Pseudonymisation module will perform a combination of techniques. The administrator of the platform will be able to define which transformations are needed to ensure proper pseudonymisation or anonymisation.

The set of transformations offered will consist of both one-way hashes and two-way encryption (possibility to encrypt and decrypt the data) as well as all the data masking techniques, except from shuffling. The reason that shuffling is excluded is because it couples data of multiple subjects. If one subject revokes consent, it is difficult to undo the transformation without affecting data corresponding to other subjects.

IV. Conclusions

This work has proposed a framework that comprises two novel security mechanisms for ensuring proper authentication and authorization when accessing resources throughout our proposed data sharing platform. Our proposed methodology aims to deliver sophisticated backbone service capabilities that will enable trusted, secure, automated, robust and controlled data transactions for food certification that aims to bring competitive advantages to all food sector businesses that demand easy, fast, and actionable access to variegating food safety data from multiple devices and in various settings. In future work,

authors would like to explore the potential of AI enabled functionalities and Deep Learning approaches concerning transfer of learning and domain adaptation that allow the transfer of knowledge through sharing learned parameters.

ACKNOWLEDGMENT

This work is co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE - INNOVATE (project code: T2EDK - 00932). Also, this work is a part of TheFSM project, that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 871703.

REFERENCES

[1] gfsi, "The coalition of action on food safety." [Online]. Available: https://mygfsi.com/

[2] K. WANG, Z. CHEN, and J. XU, "Efficient traceability system for quality and safety of agricultural products based on consortium blockchain," *Journal of Computer Applications*, vol. 39, no. 8, p. 2438, 2019.

[3] T. Devriendt, P. Borry, and M. Shabani, "Factors that influence data sharing through data sharing platforms: A qualitative study on the views and experiences of cohort holders and platform developers," *Plos one*, vol. 16, no. 7, p. e0254202, 2021.

[4] I. González-Puetate, C. Marín-Tello, and H. R. Pineda, "Agri-food safety optimized by blockchain technology," *Revista Facultad Nacional de Agronomía Medellín*, vol. 75, no. 1, pp. 9839–9851, 2022.

[5] F. Antonucci, S. Figorilli, C. Costa, F. Pallottino, L. Raso, and P. Menesatti, "A review on blockchain applications in the agri-food sector," *Journal of the Science of Food and Agriculture*, vol. 99, no. 14, pp. 6129–6138, 2019.

[6] P. N. P. K. S. R. K. P. K. Papageorgopoulos, Vergeti, "An agri-food data platform for food safety and certification," in *9th Annual Conference on Computational Science and Computational Intelligenc*. CPS, IEEE, 2022.

[7] R. K. Lomotey, S. Kumi, and R. Deters, "Data trusts as a service: Providing a platform for multi-party data sharing," *International Journal of Information Management Data Insights*, vol. 2, no. 1, p. 100075, 2022.

[8] J. Jonkman, I. Badraoui, and T. Verduijn, "Data sharing in food supply chains and the feasibility of cross-chain data platforms for added value," *Transportation Research Procedia*, vol. 67, pp. 21–30, 2022.

[9] H. Feng, X. Wang, Y. Duan, J. Zhang, and X. Zhang, "Applying blockchain technology to improve agri-food traceability: A review of development methods, benefits and challenges," *Journal of cleaner production*, vol. 260, p. 121031, 2020.

[10] S. Kumi, R. K. Lomotey, and R. Deters, "A blockchain-based platform for data management and sharing," *Procedia Computer Science*, vol. 203, pp. 95–102, 2022.

[11] A. Iftekhar and X. Cui, "Blockchain-based traceability system that ensures food safety measures to protect consumer safety and covid-19 free supply chains," *Foods*, vol. 10, no. 6, p. 1289, 2021.

[12] A. Durrant, M. Markovic, D. Matthews, D. May, G. Leontidis, and J. Enright, "How might technology rise to the challenge of data sharing in agri-food?" *Global Food Security*, vol. 28, p. 100493, 2021.

[13] A. Parecki. Oauth community site. [Online]. Available: https://oauth.net/

[14] Json web token. [Online]. Available: https://jwt.io/

[15] S. J. Muhammad, H. Chiroma, and M. Mahmud, "Cryptanalytic attacks on rivest, shamir, and adleman (rsa) cryptosystem: issues and challenges," *J Theor Appl Inf Technol*, vol. 61, no. 1, p. 2349, 2014.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24*. Springer, 2005, pp. 457–473.