ProTego
Cybersecurity Risk Mitigation Tools for Hospital and Care Centers

Whitepaper

Authors:

Farhad Aghili, Dave Singelée – KU Leuven

Eliot Salant – IBM Haifa Research Labs

Carlos Cilleruelo, Luis de-Marcos – Universidad de Alcalá

Henrique Carvalho de Resende, Johann Marquez-Barja – imec-University of Antwerp

ProTego: DATA-PROTECTION TOOLKIT REDUCING RISKS IN HOSPITALS AND CARE CENTERS

Project Nº 826284

Horizon 2020 Programme

Executive summary

Health care is an essential service that uses a great deal of sensitive personal data which has a high black market value being a lucrative target for data theft and ransomware attacks. It is clear that the necessary cybersecurity measures should be in place to ensure the protection of this medical and personal data. This is not a one-shot exercise; a broad range of data protection tools need to be integrated and combined to mitigate a large scala of cybersecurity attacks. The ProTego project has performed research on a set of advanced data protection techniques that can be used by hospital and care centers to protect the sensitive data in their systems. These techniques include data encryption, access control, end-device security and secure network communication. Within this whitepaper, the research contributions of the ProTego project in each of these domains will be discussed. Each of the proposed cybersecurity mitigation tool is aimed at preventing specific cybersecurity risks. Besides the strong security protection offered by this technology, another approach is the modular approach taken in the ProTego project. Since there are no strong dependencies between the individual components, depending on the specific needs of the end-user (e.g., hospital), one can opt to only use a subset of the available ProTego cybersecurity risk mitigation tools. Moreover, although the ProTego project specifically focused on a healthcare settings, the tools presented in this whitepaper can be applied in other settings that require data protection as well.

Keywords: Parquet Modular Encryption, Access Control, Fybrik, Continuous Authentication, Network Slicing

1. Introduction

A well known and widely adapted concept in the domain of cybersecurity is the NIST cybersecurity framework[1]. This framework is shown in Figure 1 and integrates the industry standards and best practices to help organizations manage their cybersecurity risks. It categorizes all cybersecurity processes and activities into 5 core functions:

- **Identify:** What needs to be protected?
- **Protect:** Measures to protect the necessary assets.
- **Detect:** Measures to identify cybersecurity incidents.
- **Respond:** Measures to reduce the impact of cybersecurity incidents.
- **Recover:** Measures to restore capabilities and services after a cybersecurity incident has occurred.

Figure 1. NIST cybersecurity framework.

In practice, organisations (including hospitals and care centers) need to implement all 5 categories in order to adequately address cybersecurity risks. Within the ProTego project, tools, technologies and concepts related to all these 5 categories have been developed. However, within this whitepaper, we will solely focus on the 'protect' category and perform a deep-dive into a set of data protection tools that have been studied and implemented in the ProTego project, and focus on the application of these tools within the context of healthcare (e.g., hospital or care centers). However, it is important to note that the technologies and solutions discussed in this whitepaper, can also be used in other application setings. However, this is outside the scope of this whitepaper.

[1] NIST cybersecurity framework. https://www.nist.gov/cyberframework

2. Conceptual system overview

Let us consider a typical high-level architecture of a hospital or care center from a data access point of view. This is depicted in Figure 2. Personal and medical data is being generated and needs to be securely stored in a database managed by the hospital. Only authorized entities should be able to access this data at a later stage. In this architecture, one can identify the following components:

- **Data producer:** the entity that is uploading personal or medical data to the system. This can be a medical device (e.g., IoT sensor or medical implant), the patient, or a medical staff member. We do not make any assumption on the device used by the data producer. So it could be a personal device (e.g., mobile phone, tablet or computer) of the patient or medical staff member. When a personal device is used, it is safe to assume that a medical application will be running on that personal device, and that this medical application will take care of uploading the data to the system, on behalf of the user.
- **Data consumer:** the entity that wants to download personal or medical data from the system. This can again be the patient, a medical staff member or another entity. Similarly as before, we also make no assumptions regarding the device that is used by the data consumer. So it could be a personal device (e.g., mobile phone, tablet or computer of the patient or medical staff member. When a personal device is used, it is safe to assume that a medical application will be running on that personal device, and that this medical application will take care of performing read queries to the system, on behalf of the user. I.e., all operations done by the data consumer will be done via a medical application running on the personal device.
- **Database:** this is the storage system where the personal and medical data is securely stored.This can be a storage system internally within the hospital, or a cloud-based system. Within a medical context, it is likely that the data will be stored in FHIR (Fast Healthcare Interoperability Resources) format. In Figure 2, the database is depicted on the top left in the figure.
- **Data Gateway:** the data producer or consumer will typically never connect directly to the database. Rather, they will connect to a gateway (denoted as Data Gateway) that will act as the interface to the database. The Data Gateway will send or fetch the data to/from the database, and send it to the data consumer (in case of a read operation).
- **External IAM:** it is not desired that any random user can send data to the system or request data access. Instead, the system needs to know which user wants to upload/read data from the system. Therefore, the data producer/consumer needs to be authenticated so that the system knows which user is performing this operation (e.g., patient X or doctor Y). The component used to identify and authenticate users, is the Identity and Access Management (IAM). Hospitals typically already have an IAM in place to manage their users. Therefore, within ProTego, we rely on such an external IAM to take care of this functionality. We assume that after successful authentication, the external IAM will issue a token to the data producer/consumer which can be presented to the system (to prove ones identity).

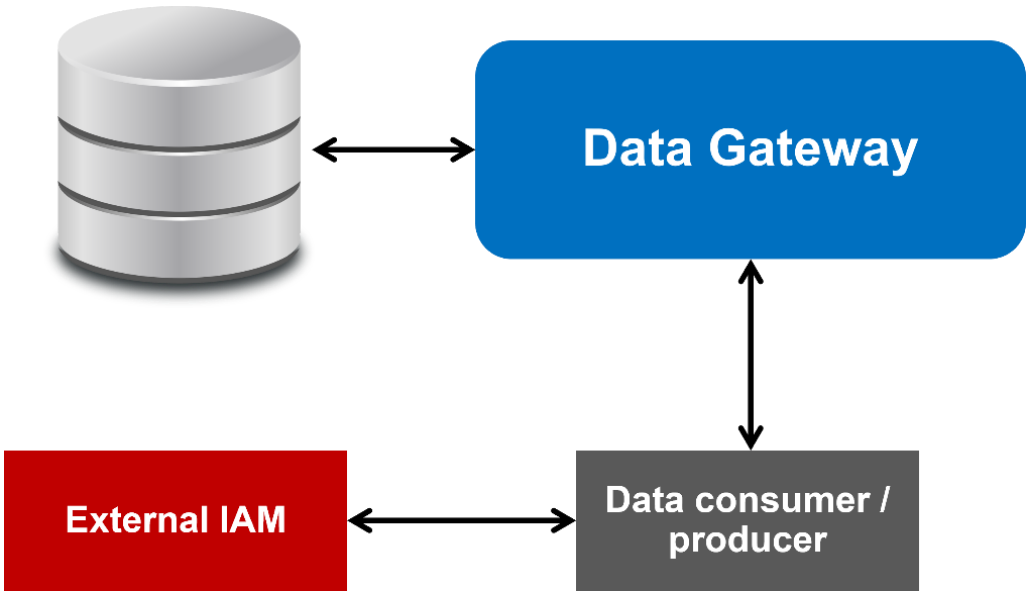

Figure 2. High-level data access architecture for hospital or care center

Let us now zoom in a bit more on the data flow. From a high-level point of view, writing data in the system works as follows. The data producer will first authenticate itself to the external IAM of the hospital. After successful authentication, it will receive a token. Next, the data producer will connect to the Data Gateway and send the data to be securely stored together with the token it received from the IAM. When the data producer is authorized to store data in the system, the Data Gateway will send the data to the database, where it securely stored.
Similarly, the reading process works as follows. The data consumer will first open the medical application on his/her personal device. When running the application, the data consumer will be asked to authenticate itself to the external IAM of the hospital. After successful authentication, it will receive a token. Next, the data consumer (via the medical application) will connect to the Data Gateway and send a query – specifying the data it wants to read – together with the token. When the data consumer is authorized to read the requested data, the Data Gateway will fetch the data from the database and send it to the mobile application running on the personal device of the data consumer.

3. Cybersecurity risks related to data protection

Considering the system overview presented in the previous section, let us now zoom in to various relevant cybersecurity risks that need to be considered, and more specifically cybersecurity risks related to data protection. The overview below is not intended to be complete, but rather focuses on a subset of important and relevant threats/risks that were tackled in the ProTego project.

Data breach in database

Sensitive data is stored in a database. This storage can be either on a physical hard disk on a computer (server), or in a cloud system. In either case, there is always the risk that somebody gets physical or remote access to the raw data (i.e., bits) stored in the storage system. An example could be an administrator responsible for managing the storage system, who obvious has physical/remote access to this storage system. But one could even imagine more stealthy or advanced attacks where an adversary succeeds in getting physical access to the storage system, and potentially even steals the hard disk(s) containing the sensitive data. In any case, if the raw data is stored in cleartext on the storage system, or if the encryption key is stored on the same physical system as the one where the encrypted data is stored (e.g., harddisk), there is clearly a data and privacy breach when an attacker succeeds in getting access to the storage system. Therefore, there is a need for strong encryption of the medical data, and secure storage and access to the encryption keys required to encrypt/decrypt the data.

Unauthorised data access by illegitimate user

Data producers and consumers connect to the hospital system and more particularly the Data Gateway via a public network connection. Therefore, any user on the Internet could also connect via this same link to try to access the sensitive medical data in the system. This unauthorised access should obviously be prevented, and shows the need to identify and authenticate users before they are able to connect to the hospital system. As mentioned before, this is realised by using an Identity and Access Management (IAM) system. Assuming the IAM system is implemented and configured correctly, only legitimate users will be able to successfully authenticate to the IAM and hence obtain a valid token.

Within the ProTego project, we rely on two commercial IAM systems – KeyCloak[2] and Amazon Cognito[3] – but obviously other IAM systems could be used as well. After having successfully authenticated to any of these IAM systems, a JSON Web Token (JWT) is produced. This JWT token contain the required information of the user (e.g., the identity of the user, the role of the user, etc.) and an expiration date. The token is digitally signed by the IAM. As a result, an illegitimate user cannot authenticate to the IAM, and cannot forge a JWT token as it cannot forge the digital signature. Therefore, by requiring a valid JWT token and checking the validity of the token, an illegitimate user cannot successfully perform any read query on the data.

An important sidenote is that in our system architecture, we have deliberately split the functionality of user management & authentication on the one hand and access control on the other hand. The former is done by the IAM, and the security components within the hospital (i.e., the access control framework, as will be discussed later) take care of the latter. In other words, the access control framework does not need to be aware of which user is part of the set of legitimate users in the system. This reduces the complexity of the overall system, which is strongly recommended from a cybersecurity point of view.

Unauthorised data access by legitimate user

Illegitimate users are prevented from unauthorised access to the sensitive medical data because these users do not get a valid token from the IAM and are not able to forge a token themselves, due to the digital signature of the token. However, there are other cybersecurity threats and risks related to unauthorised access. Indeed, also legitimate users are typically authorised to only access specific data entries, and should not be able to access any other data. The rules to decide which data access should be authorised and which data access is considered unauthorised, are denoted as *security policies*. These security policies are typically defined and enforced on a hospital level (i.e., these rules apply for all data stored and processed within the hospital, and for all users managed by the hospital). How to define these security policies are outside the scope of this whitepaper. A few practical examples that could be considered as relevant security policies for a hospital, could include:

- Patients can only access their own data
- Doctors can only access the data of their own patients

Therefore, this risk can be reformulated as follows. One needs to prevent that a legitimate user succeeds in violating the security policies in place, and successfully can retrieve data for which (s)he is not authorised for. If a user succeeds in bypassing the security policies, there is again a risk for a data breach, since sensitive data is leaked.

To prevent this risk, the access control framework should not only check the validity of the JWT token it receives from a data producer/consumer, but should also verify that the corresponding queries of that specific data producer/consumer do not violate the security policies. In other words, there is the need for an access control framework that verifies and enforces the security policies in place when a user (i.e., data consumer) performs read queries to access the medical data.

Note that due to the use of an IAM, a legitimate user cannot impersonate another user as (s)he cannot authenticate as that other user to the IAM and cannot forge a token. Therefore, that security threat is mitigated by the secure use of an IAM.

[2] https://www.keycloak.org/

[3] https://aws.amazon.com/cognito/

Personal device being lost or stolen

When the user authenticates to the external IAM, (s)he will receive a JWT token after successful authentication. This token will then be presented to the Data Gateway (and next to the access control framework) when data queries are being made. This JWT token will have an expiry date encoded in one of its fields. As long as the token has not been expired, it can directly be used to perform data queries, i.e., the user does not need to authenticate to the IAM first. Similarly, when the application that the data consumer is using – and from which the data queries are being generated – is active for a long time, one can use the current JWT token (before it expires) to authenticate in the background the user to the IAM and obtain a new fresh JWT token. By doing this, one avoids that the user has to regularly re-authenticate itself to the IAM, which would not be acceptable from a user friendliness point of view. Keeping the user authenticated for a long time of course also introduces a security risk. If the user would lose his/her personal device and the medical application would still be active on the device, or when the personal device would get stolen, there is the risk that a malicious entity would use the personal device to get access to medical data. If the JWT token on the device is still active and not yet expired, the malicious entity would not have to authenticate when trying to download/access medical data. Due to the sensitivity of the medical data, one needs to mitigate this security risk by regularly monitoring that the user of the medical application is actually the legitimate user. When this verification fails, the user of the medical application should be forced to re-authenticate itself to the IAM.

Insecure network connections

The encryption of the medical data, the secure storage (and access management) of the encryption keys, combined with an access control framework, a secure IAM and the use of continuous authentication on the personal device of the data consumer realises protection of the data at rest. Direct access to plaintext medical data is not possible, and only legitimate users can access medical data, and only the data they are authorised to access. However, one also needs to consider to secure data in transit. The medical data needs to secured when being uploaded or downloaded from the system, and when in transit between the different technical components in the ProTego architecture. Multiple security risks need to be considered here. First of all, there is the risk that sensitive data is leaked when in transit. Similarly, there is a need for data integrity and authentication to ensure that the medical data is not being manipulated when transported from one component/entity to another, and the data source (i.e., which party is sending the data) can be guaranteed. Another risk related to data in transit, is availability. One needs to ensure that critical data flows between components, particularly security-related data flows, can achieve their required quality of service requirements, such as latency or throughput. This is particularly a risk when different data flows are sent over the same physical network infrastructure. For example, one would have to avoid that if one particular service consumes a lot of bandwidth, other security-critical communication flows (e.g., fetching cryptographic keys or sending alerts to a SIEM) would no longer be possible. To mitigate these security risks related to data in transit, one needs to combine multiple mitigation strategies:

- End-to-end security by deploying Transport-Layer Security (TLS)
- Network slicing to isolate different services on the same network, both from a performance as from a security point of view.

4. Data protection tools developed within ProTego

The goal of the ProTego project is to address the cybersecurity risks mentioned in the previous section, and study and develop countermeasures to these risks. More specifically, the following risk mitigation components have been developed within ProTego:

- Apache Modular Encryption
- Access Control Framework
- Continuous Authentication
- Network slicing

Each of these technologies addresses one or more of the cybersecurity risks discussed before. In the rest of this section, we will now discuss each of these security components more in detail.

Apache Modular Encryption

FHIR is a standard describing the data formats and an API for exchanging electronic health records (EHR). The standard was created by the Health Level Seven International (HL7) health-care standards organization and has quickly gained widespread acceptance in the medical IT field. When storing and processing FHIR data, multiple challenges have to be addressed, such as efficient storage and fast processing of large amounts of FHIR resources, confidentiality and integrity of healthcare data, and efficient export of bulk FHIR data. One of the solutions to these technical challenges is the use of Apache Parquet – a popular standard for efficient storage of big, tabular data. Apache Parquet is a highly optimized format for data in tabular format (i.e., data stored into columns and rows) that enables data encoding, compression, columnar projection, etc. Using Parquet instead of traditional data formats can lead to one or two orders of magnitude speedup in workload execution[4]. Due to these properties, Apache Parquet is currently the industry-leading standard for the formatting, storage, and efficient processing of big data, and is supported by Apache Spark, an open-source unified analytics engine for large-scale data processing.

There are clearly many benefits of using Apache Parquet as one of the HL7 FHIR standards for bulk export of data as well as data storage. However, since the FHIR data is medical data and therefore sensitive data, its confidentiality and integrity needs to be protected. Therefore, within ProTego, Parquet Modular Encryption (PME) has been proposed as a solution to address the issues of protecting the confidentiality and integrity for sensitive Parquet data, in a way that won't degrade the performance of analytic systems. If the FHIR data is stored in columnar format (which will be the case when using Apache Parquet), then one can use Parquet modular encryption to encrypt sensitive columns when writing Parquet files, and decrypt these columns when reading the encrypted files. Encrypting data at the column level, enables one to decide which columns to encrypt and how to control the column access. Besides ensuring confidentiality, Parquet modular encryption also protects the integrity of stored data. Any tampering with file contents is detected and triggers a reader-side exception.

By default, Parquet modular encryption uses the AES-GCM algorithm, which is a widely used authenticated encryption algorithm, to encrypt and authenticate the raw data. However, it also supports other encryption algorithms depending on the specific security and performance requirements. Parquet modular encryption is flexible, in the sense that (i) one can use different encryption keys for different columns in the bulk FHIR data, and (ii) because it allows for partial encryption – i.e., encrypting only column(s) with sensitive data. Online, one can find instructions for developers on how to use Parquet modular encryption in the IBM product Cloud Pak for Data[5]. The code repository and a more detailed discussion on its functionality can be found on the Parquet modular encryption GitHub[6].

Access Control Framework

In ProTego, the Access Control Framework is responsible for managing the access to patients' medical data, which has been sent over a secure channel from the data producer to the Data Gateway (DGW) and stored encrypted in an external database using Parquet modular encryption. The Data Gateway uses cryptographic keys to encrypt/decrypt the medical data stored in external

[4] https://gidon-16942.medium.com/apache-parquet-for-hl7-fhir-c23610131f8c

[5] https://dataplatform.cloud.ibm.com/docs/content/wsj/spark/parquet-encryption.html?linkInPage=true&cm_sp=ibmdev-_-developer-articles-_-cloudreg

[6] https://github.com/apache/parquet-format/blob/apache-parquet-format-2.7.0/Encryption.md

storage (e.g., cloud system). The access control framework consists of three main functional components:

- (External) IAM: The IAM component is responsible for authenticating the user and managing his roles and attributes. It produces an assertion on these properties of the user that third parties can evaluate (in our case, the Access Control component). Note, the IAM is responsible for managing the attributes of all system users and generating the JWT token based on these attributes and a pre-defined expiration time.
- Key Management System (KMS): The KMS component is responsible for securely storing the encryption/decryption keys Keys it receives from the Data Gateway. In PRoTego, the open-source Vault[7] KMS is used. The secure storage of the keys within the KMS is based on an internal master key stored in the KMS. Moreover, the KMS is tight to the access control component in the following way. When the DGW receives a request from a data producer/consumer to access specific data, the access control component will evaluate this request, based on the security in place, the attributes of the user (encoded in the JWT token) and the data that was requested. When the access control component approves the access control request, it will inform the KMS to send the necessary encryption key to the DGW. If the access control request is rejected, the KMW will not send the encryption key to the DGW. In other words, the access control component acts as a gatekeeper between the DGW and the KMS.
- Access Control: The main role of this component is to evaluate access control requests from a data consumer/producers to specific medical data.Therefore, the main goal is to prevent unauthorized users from retrieving the plaintext medical data. However, besides security, it is also important that the access control component is flexible and supports more complex security policies.

Within ProTego, multiple design approaches for realising the access control component have been explored. These can be grouped into three categories and will be briefly discussed below:

- Basic access control scheme that provides support for role-based access control (RBAC) [San98] and simple static access control policies.
- Flexible access control schemes that support more dynanic and flexible access control policies.
- Access control solutions that offer advanced security protection, for example against compromises by internal adversaries.

Basic access control scheme

This scheme uses the role-based access control (RBAC) model based on some static access control rules. This is used as a baseline solution, and further improved in the other access control schemes proposed within ProTego.

Dynamic access control

The basic access control component does not allow to easily adapt the security policies, which is a disadvantage for practical systems. Therefore, two solutions have been proposed in the ProTego project to offer more flexibility.

The first approach is based on the Open Source **Fybrik** framework. Fybrik is a sophisticated system for creating a secure, locked-down path between a data source and a data consumer or producer. Conceptually, Fybrik consists of a Control Plane, which handles all the behind-the-scenes details of allowing policy-based access to a data source, and a runtime environment which allows for consumption of the data. The Control Plane includes a Policy Manager which both

[7] https://www.vaultproject.io/

allows for defining data access policy and evaluates access decisions to the data. Based on a description of the data requestor (e.g. expressed purpose of data use, role, geographic location, data source required etc.), a Fybrik Module will be deployed in the Control Plane and will enforce the access decisions. More information on Fybrik can be found in the Fybrik whitepaper[8] or the Fybrik website[9].

The second approach explored within ProTego is a **dynamic access control scheme** that relies on the Open Policy Agent (OPA)[10] framework. OPA provides a high-level declarative language (Rego) that lets us specify policy as a code. OPA works based on inputs and a Rego file. In this scheme, the data producer provides inputs, including the users' identity that can access the medical data, and the policies that the system administrators has defined and which have been transformed in a Rego file– the Rego file is located outside of the Access Control component. OPA then generates policy decisions by evaluating the query input and the security policies stored in Rego files. More information on this dynamic access control scheme can be found in the ProTego whitepaper on ProTego-ACC[11].

Attribute-based encryption access control scheme

Both the basic access control scheme as the two solutions for dynamic access control do not offer protection against advanced security threats, such as a malicious insider compromising both the DGW and access control components. Therefore, within ProTego we proposed two enhanced access control solutions: (i) the **CP-ABAC** scheme that relies on Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [BSW07] and (ii) the Cryptographic Attribute-Based Access Control (**C-ABAC**) scheme that also supports user revocation. In both schemes, the main idea is that the data is additionally encrypted by the data producer (so in addition to the encryption done in the DGW) in such a way that each data consumer that meets some pre-defined access control policies, can decrypt the data. More information on these attribute-based encryption access control schemes can be found in the ProTego whitepaper on ProTego-ACC

Continuous Authentication

Continuous Authentication (CA) is an access control mechanism that monitors the user activity to determine if access is legitimate. The Continuous Authentication component developed inside the ProTego project, focuses on providing improved security in mobile devices. Through the development of this solution, several experiments and design architectures were tested.
On one hand, an architecture capable of obtaining behavioural events from a mobile phone was designed [SCM20, MCS21]. We address this problem as a client-server situation and divided the architecture into three components:

- Authorization server: Responsible for the authentication in the architecture.
- Resources server (named JBCA API): Manages a trust value, obtained from different features of the users' identity in order to create a global trust value.
- Clients: Act as the interface between the users and the architecture. It oversees the collection of behavioural events and sends them to the resource's server.

On the other hand, it was necessary to address how to detect a behavioral pattern to calculate a trust value for each user. A CA system needs to evaluate the data received and there are multiple methods for processing the information retrieved. Our solution is based on the use of machine

[8] https://fybrik.io/dev/static/fybrik-whitepaper-2021.pdf

[9] https://fybrik.io/v0.5/

[10] https://www.openpolicyagent.org/

[11] https://protego-project.eu/wp-content/uploads/2021/12/ProTego_ACC_white_paper.pdf

learning models capable of detecting suspicious activity. In order to evaluate several algorithms and their reliability, we tested multiple algorithms using a public dataset called, H-MOG[12].
In total seven different classifiers were then trained and tested using keystroke dynamics captured from mobile devices' soft keyboard events from the HMOG dataset. The results show that ensemble algorithms (RFC, ETC, GBC) performed better, with an average accuracy of around 0.70 [MHS21]. Due to this analysis the final solution deployed in the ProTego project use ensemble algorithms as a system for computing the trust value of each user.
More details on the Continuous Authentication Solution designed in ProTego can be found in the ProTego whitepaper on continuous authentication[13].

Network slicing

The ProTego network slicing tool is an extra element of the ProTego toolkit, and is used to increase security at network level within the hospital network infrastructure. This tool allows virtualizing the available physical network elements into logical topologies. In ProTego, the Network slicing solution has two main functions: performance isolation and privacy isolation.

- Performance Isolation: provides a per-flow isolated communication channel ensuring that each use case has its own required Quality of Service (QoS). Secured flows need to have a guaranteed QoS (measured by e.g. bandwidth, latency and packet loss) to perform the required security processes.
- Privacy Isolation: adds security and data channel confidentiality to each slice, and logically separates each flow from each other, allowing per-flow topology management and per-flow encryption.

The ProTego Network Slicing tool was implemented over state-of-the-art networking tools such as 5G-EmPOWER[14], Open vSwitch, VXLAN, and IPSEC. 5G EmPOWER enabled the usage of WiFi network slicing used in ProTego for traffic prioritization of the hospitals services for patients and staff of the hospital. Network slices for such service applications were created and secured against the misuse of the hospital wifi network. Open vSwitch and VXLAN enabled the creation of separated virtual networks for each slice by distinguishing hospital critical application communication from non-critical communication. IPSEC was used to provide an extra layer of security for the hospital critical applications which were directed to a specific slice.

[12] https://www.cs.wm.edu/~qyang/hmog.html

[13] https://protego-project.eu/wp-content/uploads/2021/11/Continuous-Authentication-Whitepaper.pdf

[14] https://ieeexplore.ieee.org/document/8680665

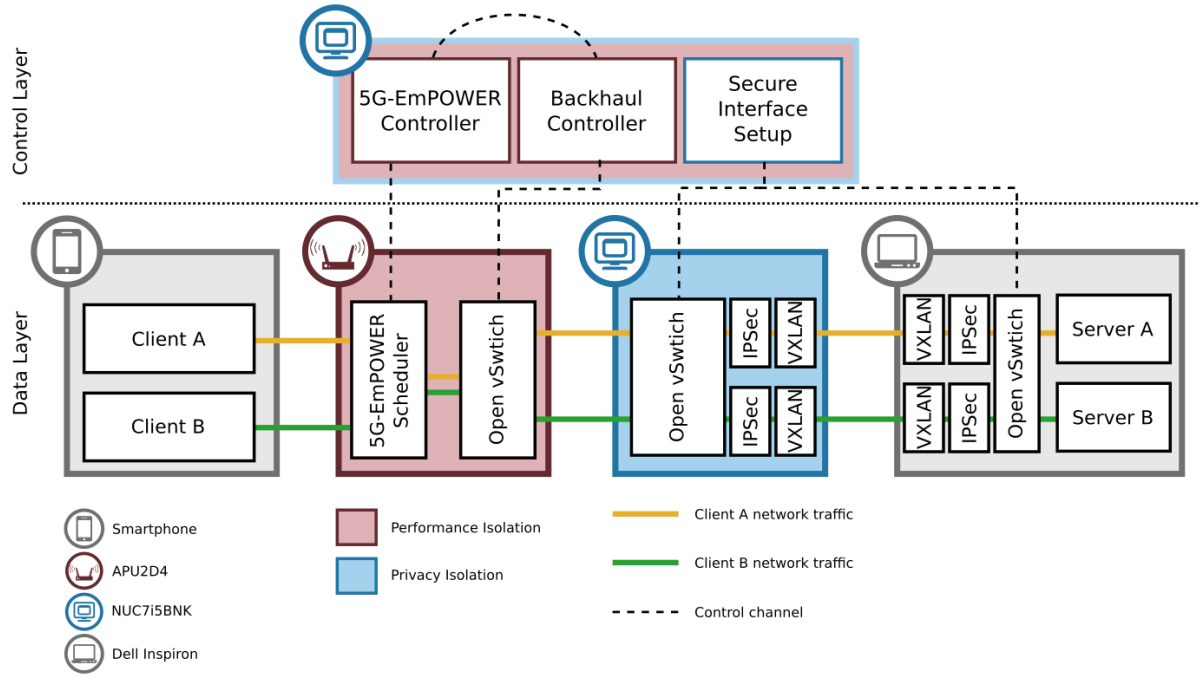

Figure 3. Network Slicing architecture

The results of our experiments [CPB20] showed that we could dynamically deploy WiFi network slices and dynamically change the traffic prioritization on client downlink OpenFlow rules using TCP and UDP protocols successfully. Furthermore, the dynamic switching of network slices, which were implemented using VXLAN, had an switch delay of less than 5 seconds, providing a feasible dynamic slice deployment on a real hospital environment. An evaluation of the trade off between performance and security caused by the extra encryption and overlay network was also assessed when deploying the network slicing tool with the partners hospitals within ProTego. Results shows that the throughput when using the network slicing privacy isolation is proportional to the processing power of the gateway and the access point used for this service. The performance isolation can be used to increase the throughput for critical hospital applications that use the privacy isolation, which can make the trade-off of having extra security layers less perceptible by end-users.

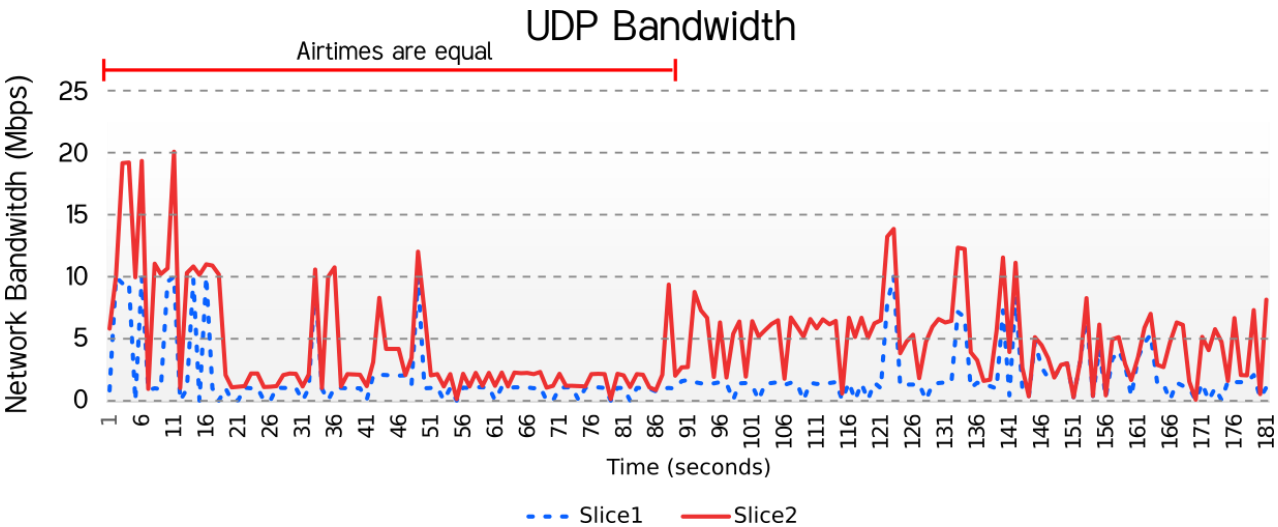

Figure 4. Bandwith when deploying Network Slicing

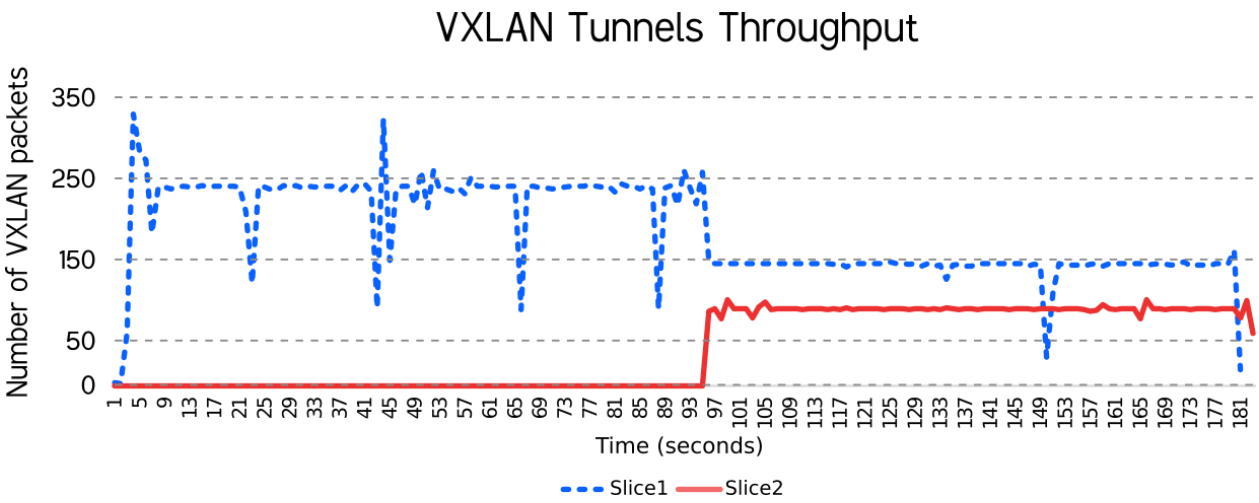

Figure 5. Throughput when deploying Network Slicing

5. Conclusion

Data protection is of uttermost importance in medical healthcare systems, for example hospitals or care centers. Security measures need to be in place to protect sensitive medical data at rest and when in transit. Within ProTego, multiple innovative cybersecurity risk mitigation tools have been developed.

Parquet files containing sensitive information can be protected by the Parquet modular encryption mechanism that encrypts and authenticates the file data and metadata - while allowing for a regular Parquet functionality (columnar projection, predicate pushdown, encoding and compression). The Parquet Modular Encryption standard has now been adopted by Apache Spark. Multiple access control solutions have been designed and developed to prevent unauthorised access and enforce fine-grained access control policies during read requests on the sensitive medical data. Continuous authentication offers mobile device security through a scalable architecture to continuously authenticate users based on their interactions with their mobile phones. Continuous authentication supports the access control functionality by preventing an adversary that has stolen the mobile phone of the legitimate user to access the sensitive medical data. Network slicing enables network isolation in the hospitals' internal networks, both from a performance and security point of view. By doing so, each network slice has a guaranteed performance. This offers performance isolation for mission critical applications in the hospitals.

One or more of these ProTego cybersecurity components can be combined with commercial-of-the-shelf building blocks, such as Identity and Access Management (IAM) systems, to offer strong data protection guarantees in healthcare infrastructures such as hospitals and care centers.

6. References

[San98] R. S. Sandhu, Role-based access control, in: Advances in computers, Vol. 46, Elsevier, 1998, pp.237–286.

[BSW07] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," 2007 IEEE Symposium on Security and Privacy (SP '07). IEEE, 2007, pp. 321-334.

[MCS21] de-Marcos, L., Cilleruelo, C., & Junquera-Sánchez, J. (2021). Mobile Continuous Authentication in eHealth: A case study for the ProTego project. Central European Conference on Information and Intelligent Systems, (pp. 325-331). Faculty of Organization and Informatics Varazdin.

[MHS21] de-Marcos, L., Martínez-Herráiz, J.-J., Junquera-Sánchez, J., & Cilleruelo, C. (2021). Comparing Machine Learning Classifiers for Continuous Authentication on Mobile Devices by Keystroke Dynamics. Electronics, 10(14), 1622.

[SCM20] Junquera-Sánchez, J., Cilleruelo-Rodríguez, C., de-Marcos, L., & Martínez-Herráiz, J.-J. (2020). JBCA: Designing an Adaptative Continuous Authentication Architecture. In Workshop of Physical Agents. Springer, 2020, pp. 194-209.

[CPB20] Henrique C. Carvalho de Resende, Joao Paulo de Brito Gonçalves, Cristiano B. Both, and Johann M. Marquez-Barja. 2020. Enabling QoS-secured Enhanced Non-Public Network Slices for Health Environments. In Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good (GoodTechs '20). ACM, 2020, pp. 18–23.