Project Acronym:	**VICINITY**
Project Full Title:	**Open virtual neighbourhood network to connect intelligent buildings and smart objects**
Grant Agreement:	**688467**
Project Duration:	**48 months (01/01/2016 - 31/12/2019)**

Deliverable D4.3

VICINITY Security Services

Work Package:	**WP4 – VICINITY Client Infrastructure Implementation**
Task(s):	**T4.3 – VICINITY Security Services**
Lead Beneficiary:	**BVR**
Due Date:	**30 June 2018 (M30)**
Submission Date:	**29 June 2018 (M30)**
Deliverable Status:	**Final**
Deliverable Type:	**DEM**
Dissemination Level:	**PU**
File Name:	**VICINITY_D4.3_Vicinity_security_services_v1_0.pdf**

VICINITY Consortium

No	Beneficiary		Country
1.	TU Kaiserslautern (Coordinator)	UNIKL	Germany
2.	ATOS SPAIN SA	ATOS	Spain
3.	Centre for Research and Technology Hellas	CERTH	Greece
4.	Aalborg University	AAU	Denmark
5.	GORENJE GOSPODINJSKI APARATI D.D.	GRN	Slovenia
6.	Hellenic Telecommunications Organization S.A.	OTE	Greece
7.	bAvenir s.r.o.	BVR	Slovakia
8.	Climate Associates Ltd	CAL	United Kingdom
9.	InterSoft A.S.	IS	Slovakia
10.	Universidad Politécnica de Madrid	UPM	Spain
11.	Gnomon Informatics S.A.	GNOMON	Greece
12.	Tiny Mesh AS	TINYM	Norway
13.	HAFENSTROM AS	ITS	Norway
14.	Enercoutim – Associação Empresarial de Energia Solar de Alcoutim	ENERC	Portugal
15.	Municipality of Pylaia-Hortiatis	MPH	Greece

Authors List

Leading Author (Editor)				
	Surname	**First Name**	**Beneficiary**	**Contact email**
	Horniak	Martin	BVR	martin.horniak@bavenir.eu
Co-authors (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Vanya	Stefan	BVR	stefan.vanya@bavenir.eu
2.	Oravec	Viktor	BVR	viktor.oravec@bavenir.eu
3.	Almela Miralles	Jorge	BVR	jorge.almela@bavenir.eu

Reviewers List

List of Reviewers (in alphabetic order)				
No	**Surname**	**First Name**	**Beneficiary**	**Contact email**
1.	Wall	Nigel	CAL	nw@nigel-wall.co.uk
2.	Samovich	Natalie	ENERC	n.samovich@enercoutim.eu
3.	Belesioti	Maria	OTE	mbelesioti@oteresearch.gr

Revision Control

Version	Date	Status	Modifications made by
0.1	31.October 2017	Initial Draft	Martin Horniak (BVR)
0.2	31. October 2017	First Draft formatted with contributions received	Jorge Almela Miralles (BVR), Viktor Oravec (BVR), Asbjørn Hovstø (HITS), Maria Belesioti (OTE)
0.2.9	7. June 2018	Update authorization mechanism	Martin Horniak (BVR)
0.3	8. June 2018	Deliverable version uploaded for Quality Check	Viktor Oravec (BVR), Martin Horniak (BVR)
0.3.2	29. June 2018	Quality Check	Viktor Oravec (BVR), Martin Horniak (BVR)
0.3.2	29. June 2018	Final Draft reviewed	Viktor Oravec (BVR), Martin Horniak (BVR)
1.0	29. June 2018	Cutting size of Executive Summary. Submission to the EC	Christoph Grimm (UNIKL)

Executive Summary

This deliverable explains the general view on cybernetic security and security goals that need to be achieved by a VICINITY system in line with security requirements and security design in D1.5 and D1.6 deliverables of the VICINITY project. Alternative definitions of what needs to be considered in regard to cyber security: we conclude that ISO/IEC 27000:2016 should be used.

The deliverable identifies following measures that are implemented into the VICINITY system and should provide reasonable protection of data that are transferred across the system:

- Authentication mechanism;
- Frontend with policy definition capabilities;
- Service and functionality for policy enforcement;
- IDS/IPS measures in platform as a service (cloud) provider;
- Strong firewall rules on all VICINITY servers;
- TDE on all core databases and slated hash password storage;
- Principle of least privilege whenever a component is deployed;
- Usage of valid certificates for all secure communication channels;
- No non-secure communication;
- Logging and audit trails collection
- Data access contracts - consents.

This deliverable focuses on the technical aspects of the defence in depth model. The Administrative aspects (training of staff, social engineering, local legislation) and physical aspects (security of the building housing the core servers) of the VICINITY security will be implemented by Pilot site demonstration teams, VICINITY Cloud, VICINITY Client infrastructure and VICINITY value-added service implementation and maintenance team. Moreover, security in IoT environment brings challenges that can't be fully addressed, some of which are discussed in the last chapter of this document.

List of Definitions and Abbreviations

Abbreviation	Definition
ACID	Atomicity, Consistency, Isolation, Durability - A set of properties of database transactions intended to guarantee validity even in the event of errors, power failures, etc. In the context of databases, a sequence of database operations that satisfies the ACID properties and, thus, can be perceived as single logical operation on the data, is called a transaction.
CA	Certificate Authority - An entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.
CCTV	Close-circuit television
CRL	Certificate Revocation List - A list of digital certificates that have been revoked by the issuing Certificate Authority before their scheduled expiration date and should no longer be trusted.
DB	Database - An organized collection of data. A relational database, more restrictively, is a collection of schemas, tables, queries, reports, views, and other elements.
EC	European Commission
EU	European Union
HTTP	Hypertext Transfer Protocol - An application protocol for distributed, collaborative, and hypermedia information systems. It is the foundation of data communication for the World Wide Web.
HTTPS	Hypertext Transfer Protocol Secure - An adaptation of the HTTP for secure communication over a computer network encrypted by Transport Layer Security (TLS), or formerly, its predecessor, Secure Sockets Layer (SSL).
IDS	Intrusion Detection System - A device or software application that monitors a network or systems for malicious activity or policy violations.
IPS	Intrusion Prevention Systems - Also known as intrusion detection and prevention systems (IDPS), are network security appliances that monitor network or system activities for malicious activity and are able to actively prevent or block intrusions that are detected.
MD	Message digest
OCSP	Online Certificate Status Protocol - An Internet protocol used for obtaining the revocation status of an X.509 digital certificate. It is described in RFC 6960 and is on the Internet standards track. It was created as an alternative to certificate revocation lists (CRL), specifically addressing certain problems associated with using CRLs in a public key infrastructure (PKI).
P2P	Peer-To-Peer - A distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes.
PKCS	Public Key Cryptography Standards - A group of public-key cryptography standards devised and published by RSA Security Inc, starting in the early

Abbreviation	Definition
	1990s. Some of the standards have begun to move into the "standards-track" processes of relevant standards organizations such as the IETF and the PKIX working-group.
PKI	Public Key Infrastructure - A set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption.
RA	Registration Authority - An entity that verifies the identity of other entities requesting their digital certificates to be stored at the CA.
RDBMS	Relational Database Management System - A database management system that is based on the relational model.
SASL	Simple Authentication and Security Layer – A framework for authentication and data security in Internet protocols.
TDE	Transparent Data Encryption - A process that uses an algorithm to transform data stored in a database into "cipher text" that is incomprehensible without first being decrypted. Transparent data encryption is used to encrypt an entire database, which therefore involves encrypting "data at rest".
TLS	Transport Layer Security - Cryptographic protocols that provide communications security over a computer network. Several versions of the protocols find widespread use in applications such as web browsing, email, Internet faxing, instant messaging, and voice-over-IP (VoIP). Websites are able to use TLS to secure all communications between their servers and web browsers. It is a successor to SSL (Secure Sockets Layer).
VPS	Virtual Private Server - A virtual machine sold as a service by an Internet hosting service. For many purposes they are functionally equivalent to a dedicated physical server, and being software-defined, are able to be much more easily created and configured.
XMPP	Extensible Messaging and Presence Protocol - A communications protocol for message-oriented middleware based on XML (Extensible Mark-up Language). Designed to be extensible, the protocol has been used also for publish-subscribe systems, signalling for VoIP, video, file transfer, gaming, the Internet of Things (IoT) applications such as the smart grid, and social networking services.

Content

1. Introduction

The purpose of this deliverable is to explain technical approach to information security measures implemented in VICINITY project. By attempting to identify vulnerabilities and threats to information resources of VICINITY system and describing countermeasures that are implemented to reduce risks to acceptable levels, it can be also taken as a part of risk management process. The first part of this document, describes general objectives of information security, along with standard approaches and key concepts used to achieve them. Information used in the first part is gathered from publicly available sources or own expertise of the implementation team. The second part is dedicated to technical description of security measures that are implemented in VICINITY, their interaction and their contribution to overall security of the system. All tasks that need to be done to achieve security goals are listed in line. Lastly, every security architecture is limited in its scope and this is especially true for very heterogenous systems, such as VICINITY. Limitations of implemented technical solutions is therefore discussed.

In general, information security is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. When discussing computer system security, the chief area of concern is the balanced protection of the Confidentiality, Integrity and Availability of data, also known as the CIA Triad, while maintaining a focus on efficient policy implementation and no major hampering of organization productivity. In addition, we shall take steps to assure Non-repudiation, as suggested by ISO/IEC 27000:2016.

The critical technical security goals within the scope of a VICINITY system must be achieved on multiple levels of VICINITY components' interaction in order to achieve real defense in depth and to ensure that data are protected during the entire life-cycle of information transport across the VICINITY system. Since there are parts of the VICINITY system where the security cannot be guaranteed by any practical measures, a zone of guaranteed security and its boundaries were demarcated as one of the first steps. It is important to understand that VICINITY does not provide end-to-end security. Only the core elements are protected so that the data sharing rules set by the data subject are fully protected. Information gathered from connected sensors may be vulnerable to breaches in confidentiality, accuracy and availability. However, it is assumed that data passed from sensors to the VICINITY gateway are assumed to be unencrypted and adequately protected.

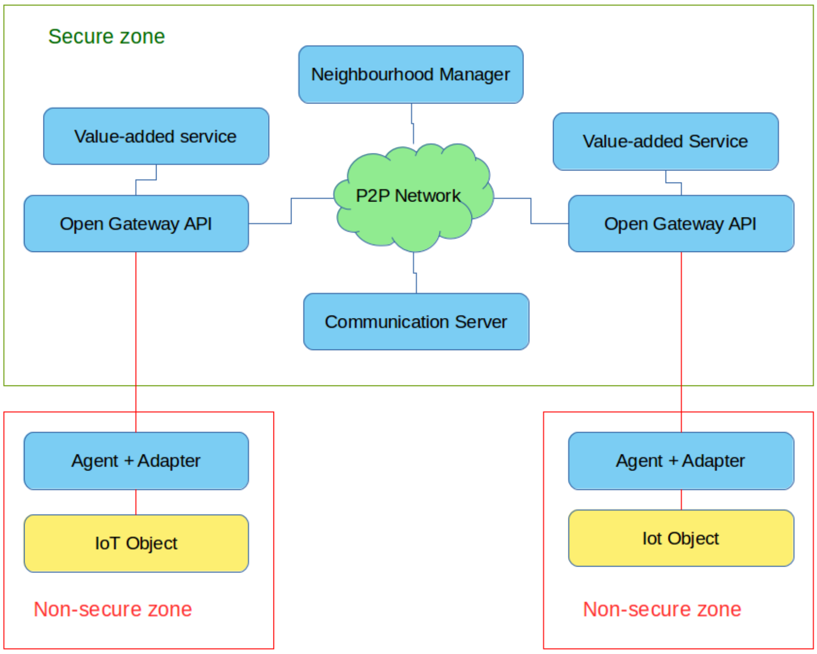

Figure 1 Demarcation of secure and non-secure zones of VICINITY system. Security is only guaranteed inside the "Secure zone".

1.1. Context within VICINITY

The D4.3 VICINITY Security services is part of WP4 Clint Infrastructure Implementation work package (Figure 2). The D4.3 is from the 3 main deliverables D1.5 VICINITY technical requirements specification, D1.6 VICINITY architecture design and D2.1 Analysis of Standardisation Context and Recommendations for Standards Involvement from which VICINITY functional, non-function requirements, design decision and standard followed has been implemented.

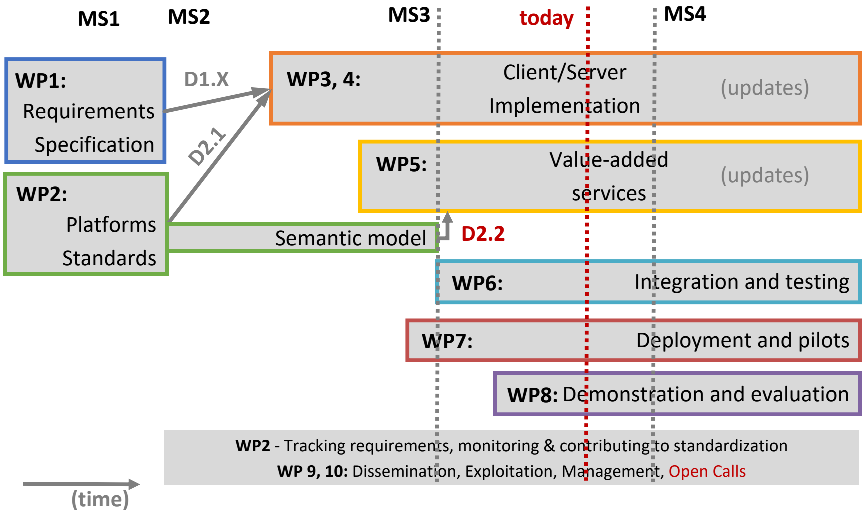
MS1
MS2
MS3
today
MS4
WP1: Requirements Specification
D1.X
D2.1
WP3, 4: Client/Server Implementation
(updates)
WP5: Value-added services
(updates)
WP2: Platforms Standards
Semantic model
D2.2
WP6: Integration and testing
WP7: Deployment and pilots
WP8: Demonstration and evaluation
WP2 - Tracking requirements, monitoring & contributing to standardization
WP 9, 10: Dissemination, Exploitation, Management, Open Calls
(time)

Figure 2 VICINITY Work package structure

1.2. Objectives in Work Package 4 and Task 4.3

The VICINITY Security services deliverable is delivered under Task 4.3 of Work Package 4 and addresses the objective:

- Objective 3.4 Advanced trust, security and privacy assuring mechanisms implemented.

To achieve this Objective 3.4 the following VICINITY Security architecture was defined and following security services were implemented:

- XMPP SASL authentication mechanism;
- Frontend with policy definition capabilities;
- Service for policy enforcement;
- Functionality for policy enforcement;
- IDS/IPS measures in place and active in platform as a service (cloud) provider;
- Strong firewall rules on all VICINITY servers;
- TDE on all core databases;
- Salted hash password storage on all applications that require manipulation with passwords;
- Principle of least privilege whenever a component is deployed;
- Usage of valid certificates for all secure communication channels;
- Disable non-secure communication channels on all services;
- Termination of communication when certificate verification or TLS handshake fails;
- Data owned consents to process data by service, through data access contracts.

2. Information security

In general, information security is the practice of preventing unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information. It is a universal term that can be used regardless of the form the data may take.

Because the information security is such a multidisciplinary area of study and professional activity, its precise definition varies among sources, being it text books, standards or local laws. However when discussing computer system security, it is widely assumed that chief area of concern is the balanced protection of the Confidentiality, Integrity and Availability of data, also known as the CIA Triad, while maintaining a focus on efficient policy implementation and no major hampering of organization productivity.[1] The members of this classic CIA Triad are interchangeably referred to in the literature as security attributes, properties, security goals, fundamental aspects, information criteria, critical information characteristics and basic building blocks.

There is however continuous discussion about extending the classic CIA Triad and other attributes have sometimes been proposed for addition. A very good example can be provided by ISO/IEC 27000:2016, which defines the information security as "Preservation of confidentiality, integrity and availability of information. Note: In addition, other properties, such as authenticity, accountability, non-repudiation and reliability can also be involved."[2] This is the definition that provides a less general approach than the classic CIA Triad, while being less complex than e.g. the security management standard O-ISM3 published by Open Group (which has 17 security objectives). [3] In the opinion of the VICINITY implementation team the ISO definition is therefore closest to optimal, when the project's scope is taken into account, and will be considered as a reference throughout the rest of this deliverable.

One more concept of security of information systems is so called defence in depth. This concept is independent on aforementioned definitions of security goals and is should be taken as complementary in its nature, by defining hierarchical levels on which the security goals should be achieved.

2.1. Confidentiality

As defined by ISO/IEC 27000:2016, confidentiality "is the property, that information is not made available or disclosed to unauthorized individuals, entities, or processes". Obviously, access must be restricted to those authorized to view the data in question.

Data encryption is a common method of ensuring confidentiality. User IDs and passwords constitute a standard procedure; two-factor authentication is becoming the norm. Other options include biometric verification and security tokens, key fobs or soft tokens. In addition, users can take precautions to minimize the number of places where the information appears and the number of times it is actually transmitted to complete a required transaction. [4]

2.2. Integrity

In information security, data integrity means maintaining and assuring the accuracy and completeness of data over its entire life-cycle. This means that data cannot be modified in an unauthorized or undetected manner. Data integrity is the opposite of data corruption, which is a form of data loss. The overall intent of any data integrity technique is the same: ensure data is recorded exactly as intended (such as a database correctly rejecting mutually exclusive possibilities,) and upon later retrieval, ensure the data is the same as it was when it was originally recorded. In short, data integrity aims to prevent unauthorised and unintentional changes to information. This is not the same thing as referential integrity in databases, although it can be viewed as a special case of consistency as understood in the classic ACID model of transaction processing.

Measures to ensure the integrity of data include file permissions and user access controls. Version control may be used to prevent erroneous changes or accidental deletion by authorized users. Some data might include cryptographic checksums for verification of integrity. Backups or redundancies can restore the affected data to its correct state.

2.3. Availability

For any information system to serve its purpose, the information must be available when it is needed. This means that the computing systems used to store and process the information, the security controls used to protect it, and the communication channels used to access it must be functioning correctly. High availability systems aim to remain available at all times, preventing service disruptions due to power outages, hardware failures, and system upgrades. Ensuring availability also involves preventing denial-of-service attacks, such as a flood of incoming messages to the target system essentially forcing it to shut down.

2.4. Non-repudiation

In law, non-repudiation implies one's intention to fulfil their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction.

It is important to note that while technology such as cryptographic systems can assist in non-repudiation efforts, the concept is at its core a legal concept transcending the realm of technology. Regarding digital security, the cryptological meaning and application of non-repudiation shifts to mean:

- A service that provides proof of the integrity and origin of data.
- An authentication that can be asserted to be genuine with high assurance.

Proof of data integrity is typically the easiest of these requirements to accomplish. A data hash, such as SHA2, is usually sufficient to establish that the likelihood of data being undetectably changed is extremely low. Even with this safeguard, it is still possible to tamper with data in transit, either through a man-in-the-middle attack. Due to this flaw, data integrity is best asserted when the recipient already possesses the necessary verification information.

The most common method of asserting the digital origin of data is through digital certificates, a form of public key infrastructure to which digital signatures belong. Note that the public key scheme is not used for encryption in this form; i.e. the goal is not to achieve confidentiality, since a message signed with a private key can be read by anyone using the public key. Verifying the digital origin means that the signed data can be, with reasonable certainty, trusted to be from somebody who possesses the private key corresponding to the signing certificate. If the key is not properly safeguarded by the original owner, digital forgery can become a major concern.

2.5. Defence in depth

Defence in depth is originally a military strategy that seeks to delay rather than prevent the advance of an attacker by yielding space to buy time. The placement of protection mechanisms, procedures and policies are intended to increase the dependability of an IT system, where multiple layers of defence prevent espionage and direct attacks against critical systems. In terms of computer network defence, defence in depth measures should not only prevent security breaches but also buy an organization time to detect and respond to an attack and so reduce and mitigate the consequences of a breach.

The idea behind the defence in depth approach is to defend a system against any particular attack using several independent methods. [5] It is a layering tactic, conceived by the National Security Agency (NSA) as a comprehensive approach to information and electronic security. [6]

Defence in depth can be divided into three areas: [7]

- administrative,
- physical,

- technical.

Administrative controls are an organization's policies and procedures. Their purpose is to ensure that there is proper guidance available in regard to security and that regulations are met. They include things such as hiring practices, data handling procedures, and security requirements.

The Physical area is anything that physically limits or prevents access to IT systems. Fences, guards, dogs, and CCTV systems and the like.

The Technical area represents hardware or software whose purpose is to protect systems and resources. Examples of technical controls would be data encryption, authentication methods, firewalls. The technical area differs from physical in that the controls prevent access to the contents of a system, but not the physical systems themselves.

Since the VICINITY project is elaborated by several partners from all around Europe, various legislative, insurance and internal rules apply to those two areas. Although interesting, both these areas are out of scope of this document. This deliverable further discusses only technical area of this concept, which is the only area where the implementation team has overall control.

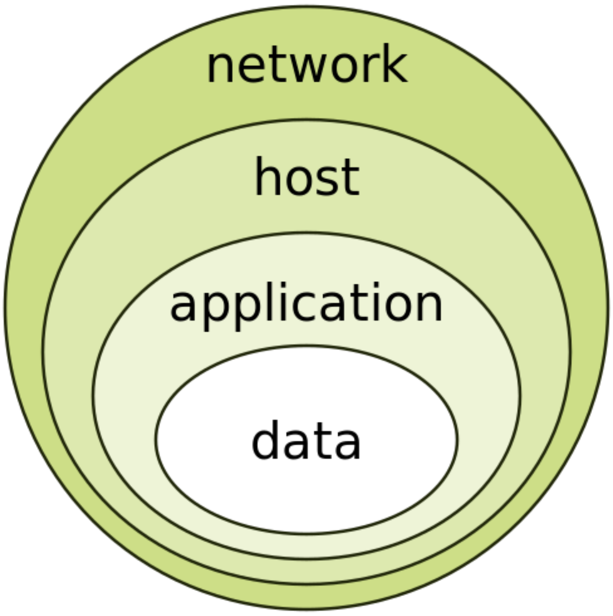

Figure 3 A graphical representation of the onion model of technical area of defence in depth. [8]

3. VICINITY Security architecture

This chapter presents the technical approach of VICINITY system to achieve security goals defined in the previous chapter and is divided into two main sections.

The first one defines what technical measure covers which security goal. Achieving a security goal is usually not possible by implementing just one feature and that comprehensive coverage often requires implementation of multiple measures or features. A good example can be the question of system availability, where ensuring just a redundant physical disk storage will certainly not protect a VICINITY core component from other factors that affect availability, such as the possibility of DoS attack. On the other hand, there are security measures that, at least partially, cover multiple security goals. Such is the case of authenticated encryption, which is a partial solution to confidentiality, integrity and data availability.

The second section describes technical details of each feature that will be implemented. Naturally, any details that would directly jeopardise the security of the system are not mentioned.

3.1. Achieving security goals

The critical technical security goals within the scope of the VICINITY system must be achieved on multiple levels of VICINITY components' interaction in order to achieve real defence in depth and to ensure that data are protected during the entire life-cycle of information transport across the VICINITY system.

It is also very important to define where are the boundaries of such security guaranteed by the VICINITY system. The VICINITY system consists of several interacting components, most of which are implemented by consortium partners. It can be then assumed that their security will be verified by some kind of (at least) internal security audit and the security of information that is processed by these components can be considered as guaranteed, in part due to legal obligations of VICINITY partners to end user agreement. The end points however are VICINITY adapters that, as defined in the VICINITY system architecture, can be implemented by a vendor of the particular smart device, which is an IoT object that is connected into the system.

As this adapter can be proprietary in nature (because the vendor can try to hide characteristics of his device's API), or the device that is running the adapter might not have enough processing power to utilize secure communication with VICINITY Open Gateway API, it is impossible for the VICINITY system to guarantee secure transfer and processing of the information up to the point of being processed by Gateway API. Therefore, guaranteed security measures only apply to the components of the system inside the "Secure zone" (see figure 3).

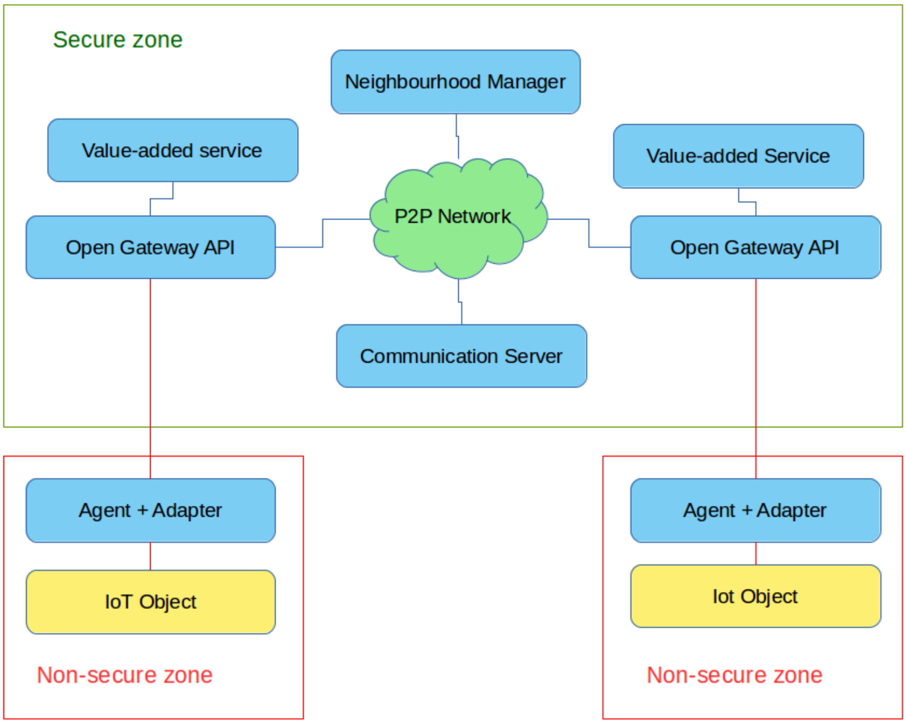

Figure 4 Demarcation of secure and non-secure zones of VICINITY system. Security is only guaranteed inside the "Secure zone".

Table 1 includes security measures implemented into the VICINITY system, and their relative overlap in solving security goals on various levels of defence in depth structure. Every security goal is covered by at least one measure on every level the defence in depth structure.

Table 1 List of main technical security measures identified as critical and implemented into the VICINITY system

Implemented measure	What is solved by implemented measure?				At what level it is solved?		
	Confidentiality	Integrity	Availability	Non-repudiation	Net-work	Host	Application
Authentication SASL	Yes	Yes	Yes	Yes	Yes	No	No
Authorization	Yes	No	No	Yes	No	No	Yes
Database encryption and password storage	Yes	No	No	No	No	Yes	No
Secure communication channels with core components	Yes	Yes	Yes	Yes	Yes	No	No
Principle of the least privilege	No	Yes	Yes	No	No	Yes	No
Firewalls, IDS/IPS	No	No	Yes	Yes	Yes	Yes	No
Comprehensive log analysis	No	Yes	Yes	No	Yes	Yes	Yes
Deployment of VICINITY components on cloud services	No	Yes	Yes	No	Yes	Yes	Yes

3.2. Detailed technical description of security measures

The following section describes the main technical security measures that are implemented into the VICINITY system, in alphabetic order. It also tries to explain how a particular security measure is affecting various security goals on multiple levels. Necessary implementation/deployment tasks are listed after each description.

The VICINITY Security architecture relies on standard commercial off-the-shelf solutions, however they are following key differentiators:

- Access to device data/ controls and events are defined directly by data owner in VICINITY Neighbourhood Manager through data access contracts - consents;
- Access to the data defined by data owner is enforced on VICINITY Open Gateway API, preferably in data owners' infrastructure.

The authentication mechanisms and end-to-end encryption implementation is divided into two phases, depending on the engine used for communication among gateways[1]. The first phase considers using XMPP protocol communication, which uses the SASL model of authentication with the Communication Server. In the second phase of VICINITY Gateway API future development, where XMPP Protocol engine would be replaced by other genuine P2P engine, and new challenges will need to be addressed. The authentication of this future version of Open Gateway will be solved by implementing three mechanisms known as Authenticated encryption, Public key infrastructure scheme and OCSP (Online Certificate Status Protocol) (see Section 3.2.9).

3.2.1. Authentication SASL

During the first phase of operation, the Open Gateway utilizes the XMPP protocol for communication. The authentication in XMPP is provided by the SASL layer.

[1] Communication engine will be visible in VICINITY Neighbourhood manager in Gateway list and in configuration of the VICINITY Gateway API.

SASL (Simple Authentication and Security Layer), is a framework for authentication and data security in several different Internet protocols. It decouples authentication mechanisms from application protocols, in theory allowing any authentication mechanism supported by SASL to be used in any application protocol that uses SASL. Authentication mechanisms of SASL also provide a data security layer offering data integrity and data confidentiality services.
SASL is defined in the RFC4422 and its implementation for XMPP protocol is described in the XEP-0034. Both documents are publicly available.

Table 2 List of tasks necessary for implementation of Authenticated encryption and PKI.

Authenticated encryption and PKI task list	**Affected VICINITY components**		
Task name	**Open Gateway API**	**Communication Server**	**Neighbourhood Manager**
Implement XMPP SASL authentication mechanism	Yes	No	No

3.2.2.Authorization

Authorization is the function of specifying access rights/privileges to resources related to information security and computer security in general and to access control in particular. [11] Access control in computer systems and networks relies on access policies. The access control process can be divided into two phases: policy definition phase where access is authorized, and policy enforcement phase where access requests are approved or disapproved. Authorization is the function of the policy definition phase which precedes the policy enforcement phase where access requests are approved or disapproved based on the previously defined authorizations.

In a VICINITY system, the policy definition phase revolves around VICINITY Neighbourhood Manager component. This component will facilitate creating, modifying or deleting access rights for each end point akin to modern social network interaction.

The policy enforcement stage will rely on VICINITY Communication Server, which will also host various components necessary for proper functioning of PKI. Every time an access policy definition is created, modified or deleted in the VICINITY Neighbourhood Manager, the VICINITY Communication Server will send a P2P message to end points that are involved in the change. After receiving the message, the end-points will react by connecting to a dedicated service over HTTPS, from where they download updated list of access policies. The enforcement action can then be implemented on either or both sides (sending or/and receiving) of the P2P communication by following approaches:

- An end-point that would like to initialize the communication (the sending side) may or may not have access to receiving side's public key (without the public key, the initial cryptographic exchange is not possible and communication cannot start).
- An end-point that would be the receiving end-point may or may not find the sender in its list of accepted peers (and will therefore refuse to communicate with a peer that is not included in the list). This can also enforce the policy on the sender side, which may refuse to even initialize communication with a peer not involved in the list.

In the first case, the mentioned dedicated service would provide a list of public keys of only those peers, that authorized the communication with the end point in question. In the second case the dedicated service would provide a list of those peers, that authorized the communication. Although the second approach might be sufficient, implementation of both rapidly increases the level of policy enforcement ability.

Table 3 contains a list of tasks that have to be done in order for this security measure to function properly.

Table 3 List of tasks necessary to achieve correct functionality of Authorization.

Authorization task list	Affected VICINITY components		
Task name	**Open Gateway API**	**Communication Server**	**Neighbourhood Manager**
Implement frontend with policy definition capabilities	No	No	Yes
Implement service for policy enforcement	No	Yes	No
Implement functionality for policy enforcement	Yes	No	No

3.2.3. Comprehensive log analysis

One of the principal defences against any cybernetic attack on an IT system is the ability of the system to recognize unexpected or malicious activity, and report and store records with as much details as possible. No measures can be taken to prevent further attacks, if the previous were not detected. And it is not just cybernetic attacks and their details that need to be recorded for forensic purposes. Simple system states that can indicate malfunctions, excessive loads on computational resources and other information important for smooth system operation need to be recorded into system logs as well.

It is obvious that the amount of such data can quickly rise in size and becomes hard for system operators to watch efficiently, making log analysis (also called audit trail) automation necessary. Technologies exists to support advanced functions of audit trails:

- Pattern recognition,
- Normalization,
- Classification and tagging,
- Correlation analysis,
- Artificial ignorance.

Pattern recognition is a function of selecting incoming messages and compare with pattern book in order to filter or handle different way.

Normalization is the function of converting message parts to same format (e.g. common date format or normalized IP address).

Classification and tagging is ordering messages into different classes or tagging them with different keywords for later usage (e.g. filtering or display).

Correlation analysis is a technology of collecting messages from different systems and finding all the messages belonging to one single event (e.g., messages generated by malicious activity on different systems: network devices, firewalls, servers, etc.). It is usually connected with alerting systems.

Artificial Ignorance a type of machine learning which is a process of discarding log entries which are known to be uninteresting. Artificial ignorance is a method to detect the anomalies in a working system. In log analysis, this means recognizing and ignoring the regular, common log messages that result from the normal operation of the system, and therefore are not too interesting. However, new messages that have not appeared in the logs before can signal important events, and should be therefore investigated. In addition to anomalies, the algorithm will identify common events that did not occur. For example, a system update that runs every week, and one week it was not run.

A software capable of some of these functions needs to be chosen and configured to audit VICINITY system logs in order to provide effective detection of intrusions and other system problems and states.

Moreover, the VICINITY Open Gateway APIs should not be omitted when discussing logging mechanisms. Although it is not viable to set up the audit trail software on machines running the VICINITY Open Gateway APIs (it is presumed that the devices might not always have the necessary computational power for such software), it is definitely viable to implement a logging capabilities on them (to a database hosted on VICINITY Communication Server, for example). This will not only make all logs available for the audit software to analyse, it will also make log inspection more comfortable for human operator (e.g. from the VICINITY Neighbourhood Manager).

For the considered VICINITY TRL the audit trail mechanism in VICINITY Neighbourhood Manager and VICINITY Open Gateway APIs logging mechanism were implemented.

Table 4 contains a list of tasks that have to be done in order for this security measure to function properly in fully operational environment.

Table 4 List of tasks necessary to employ effective audit trailing

Comprehensive log analysis	**Affected VICINITY components**		
Task name	**Open Gateway API**	**Communication Server**	**Neighbourhood Manager**
Audit trail mechanism in VICNITY Neighbourhood Manager	No	Yes	Yes
VICINITY Gateway API logging mechanism	Yes	No	No

3.2.4. Database encryption and password storage

Encryption of data that get stored on VICINITY servers contribute significantly to overall system security. Although it is rather rare that cloud service providers offer encryption of the whole virtual partition that stores a VPS (that would be potentially hosting VICINITY core components), it is still possible to configure at least an encryption of the particular database (or its part), that the component would use. Such methods of encryption are readily available for most RDBMS. There are two main approaches to DB encryption:

- Transparent/External database encryption,
- Column-level encryption.

Transparent data encryption (TDE) is used to encrypt an entire database, which therefore involves encrypting "data at rest".[12] Data at rest can generally be defined as "inactive" data that is not currently being edited or pushed across a network. As an example, a text file stored on a computer is "at rest" until it is opened and edited. TDE ensures that the data on physical storage media cannot be read by malicious individuals that may have the intention to steal them.

Perhaps the most important strength that is attributed to TDE is its transparency. Given that TDE encrypts all data it can be said that no applications need to be altered in order for TDE to run correctly. It is important to note that TDE encrypts the entirety of the database as well as backups of the database. The contents of the database are encrypted using a symmetric key that is often referred to as a "database encryption key".

Whilst TDE usually encrypts an entire database, column-level encryption allows for individual columns within a database to be encrypted. It is important to establish that the granularity of column-level encryption causes specific strengths and weaknesses to arise when compared to encrypting an entire database. Although being a very flexible approach, the main disadvantage associated with column-level database encryption is speed, or a loss thereof. Encrypting separate columns with different unique keys in the same database can cause database performance to decrease, and additionally also decreases the speed at which the contents of the database can be indexed or searched. It is therefore more viable for the purpose of the VICINITY project to utilize transparent database encryption.

One more consideration that needs to be taken into account is the storage of passwords in the database. Storing passwords in plain text will expose them to any attack, that will exploit a vulnerability of the VICINITY application - even if the database is encrypted, since the TDE protects only the "data at rest". Therefore, the passwords (or any kind of their authentication equivalents) need to be stored as hashes with unique salt for each record.

Salted hash prepends (or appends) a long pseudo random string to inserted plaintext password. The resulting string is then hashed with an algorithm that makes reverse translation non-viable using current technology. This approach limits the danger posed by any kind of rainbow table attack and must be implemented by any VICINITY application that authenticates a user or a device on the network against its stored password.

Table 5 contains a list of tasks that have to be done in order for this security measure to function properly.

Table 5 List of necessary tasks related to Database encryption and password storage.

Database encryption and password storage	**Affected VICINITY components**		
Task name	**Open Gateway API**	**Communication Server**	**Neighbourhood Manager**
Configure TDE on all core databases	No	Yes	Yes
Implement salted hash password storage on all applications that require manipulation with passwords	No	Yes	Yes

3.2.5. Deployment of VICINITY components on cloud services

Core VICINITY components (Communication Server, Neighbourhood Manager, ...) need to be deployed to a public server environment in order to be reachable for all peers in P2P network. This deployment environment can either be a dedicated high-end computer infrastructure provided by VICINITY project partners, or a contracted cloud service. In both cases, the environment needs to deliver high availability of the deployed VICINITY services.

Dedicated computer infrastructure provided by a VICINITY partner means, that all servers, network infrastructure and supporting devices and services will be provided by one of the partners involved in the VICINITY project, housed on his or her property. Contrastingly, utilization of cloud services (to be more specific, the Platform as a service model) would mean that the whole infrastructure is contracted from a third-party provider and offered as a service to the VICINITY consortium.

Having a dedicated infrastructure provided by the VICINITY partners brings some advantages, mainly from the privacy point of view. All data that would be for any reason stored on servers that are "in-house", will be a property of the partner that hosts the infrastructure (or a property of the whole VICINITY consortium, depending on agreements and local legislation) and no third party will during normal operation have access to the hardware or software. This cannot be said about Platform as a service model, where the access to the infrastructure and physical hardware is in fact beyond the control of VICINITY consortium.

Despite this apparent disadvantage, deploying the core components into the cloud still brings advantages that outweighs the disadvantages. The point is that providing a reasonably reliable dedicated infrastructure that delivers high availability, needs heavy investments from a VICINITY partner. The infrastructure will have to provide very fast Internet connection. This connection needs to have a redundant back up, which calls for various expensive active network devices. The power needs to be backed up as well (by power generators / batteries). Any physical disk arrays and servers need to have the ability to be hot swapped and data being regularly backed up to geographically distant location. Such a huge investment requires some sort of insurance, which is usually not possible without expensive fire protection system and vast security measures in place. The topic of having a staff to man it, is not even touched. On the other hand, cloud computing achieves all these requirements for availability, while remarkably cutting the regular monthly costs, thanks to the economy of scale. The Cloud service must also fully comply with GDPR.

Table 6 contains a list of tasks that have to be done in order for this security measure to function properly.

Table 6 List of tasks for Cloud deployment.

Deployment of VICINITY components on cloud services	**Affected VICINITY components**		
Task name	**Open Gateway API**	**Communication Server**	**Neighbourhood Manager**
Deploy VICINITY core components into cloud	No	Yes	Yes

3.2.6. Network and host-based firewalls, network IDS/IPS

Firewalls, IDS and IPS are standard measures serving to protect critical infrastructure from certain cybernetic attacks. A firewall looks outwardly for intrusions and limits access between networks to prevent intrusion. An IDS evaluates a suspected intrusion once it has taken place and signals an alarm. An IDS also watches for attacks that originate from within a system. This is traditionally achieved by examining network communications, identifying heuristics and patterns (often known as signatures) of common computer attacks, and taking action to alert operators. A system that terminates connections is called an intrusion prevention system (IPS), and is another form of an application layer firewall.

It is a good practice to set up a firewall on every server, being it a physical or virtual machine. Only ports, that really are used should be left open and specific critical ports (like SSH) need to have a rule to introduce a delay after an incorrect password is provided. It is possible to achieve this with a basic Iptables firewall (this serves as prevention for DoS and DDoS attacks). This approach will be taken on every server that will host a VICINITY component. However, setting up IDS/IPS is a more complex task, requiring a detailed knowledge of the underlying network and other hosts that are connected. Such configuration is nearly impossible to guess when using Platform as a service cloud model.

Fortunately, cloud services often include an IDS/IPS protection, and although it cannot be said that every provider runs these protection services, a provider that demonstrably run them, can be chosen from the vast amount of options.

Table 7 contains a list of tasks that have to be done in order for this security measure to function properly.

Table 7 List of tasks necessary to ensure that servers and underlying network infrastructure are protected.

Network and host-based firewalls, network IDS/IPS	**Affected VICINITY components**

Task name	Open Gateway API	Communication Server	Neighbourhood Manager
Verify that Platform as a service (cloud) provider has the IDS/IPS measures in place and active, and is GDPR compliant	No	Yes	Yes
Set up strong firewall rules on all VICINITY servers	No	Yes	Yes

3.2.7. Principle of the least privilege

The principle of least privilege requires that an individual, program or system process is not granted any more access privileges than are necessary to perform the task. In VICINITY Neighbourhood Manager System integrator has right to setup VICINITY Gateway for the particular IoT infrastructure, however he is not allowed to register devices or services. The principle of least privilege is widely recognized as an important design consideration in enhancing the protection of data and functionality from faults and malicious behaviour.

Table 8 contains a list of tasks that have to be done in order for this security measure to function properly.

Table 8 List of necessary tasks to correctly implement the principle.

Principle of the least privilege	Affected VICINITY components		
Task name	Open Gateway API	Communication Server	Neighbourhood Manager
Follow the principle of the least privilege whenever a component is deployed	Yes	Yes	Yes

3.2.8. Secure communication channels with core components

VICINITY Open Gateway API will communicate not only on P2P network, it will also utilize HTTP to obtain vital data from VICINITY core components, or to send logs to central storage. Since the HTTP is not encrypted and sensitive data are expected to be transferred over the network, its usage (as well as any other unencrypted protocol) is strongly discouraged and HTTPS is proposed to be used instead (in case of other used protocols, they should be protected by TLS).

In order for these secure channels to work properly, certificates signed by proper Certificate Authority need to be put on the servers running the core components and their utilization need to be correctly configured for all services the servers will provide. Unencrypted protocols should be disabled.

Moreover, the VICINITY Open Gateway API needs to be implemented the way that it refuses to proceed with communication when server's certificate validity cannot be verified, and the TLS handshake fails.

Table 9 contains a list of tasks that must be done for this security measure to function properly.

Table 9 List of tasks necessary to achieve secure communication.

Secure communication channels with core components	Affected VICINITY components

Task name	Open Gateway API	Communication Server	Neighbourhood Manager
Configure usage of valid certificates for all secure communication channels	No	Yes	Yes
Disable non-secure communication channels on all services	No	Yes	Yes
Implement termination of communication when certificate verification or TLS handshake fails	Yes	No	No

3.2.9. Privacy

As defined in D1.6 VICINITY Architecture, VICINITY shall rely on privacy impact assessments of organization and service provided. Result of the privacy impact shall be included in organization and service profiles.

VICINITY user is able to provide and revoke authorization of data access contract – data processing consent - produced by his device and services at any time using VICINITY Neighbourhood manager. The authorization is approved by both sides data owner or data controller.

Moreover, VICINITY is not storing any user data from integrated infrastructure, devices and service, except data needed to perform services provided by VICINITY. Any private data stored in user, organization, device and service profiles can be updated by user and removed when not used any more if possible (device removed, service removed, user sign out, organization sign out). Provision of these data should be covered by terms and conditions. Note, that in fully operational mode of VICINITY legal constraints can apply in case of data erasure.

Privacy	Affected VICINITY components		
Task name	Open Gateway API	Communication Server	Neighbourhood Manager
Implement data access contracts - consents	No	Yes	Yes
Implement device, service, organization and user profiles	No	No	Yes

3. Future VICINITY Security services

While the VICINITY Security architecture needs to be periodically evaluated then it is worth to specify security services which will not rely on XMPP security features such as SASL Authentication and might be implemented in case of security requirements introduced by VICNITY Platform promoting to operation level. In such case Authenticated encryption and PKI infrastructure should be considered to be configured in VICINITY Architecture. The following chapter discusses this topic in more details.

3.2.10. Authenticated encryption and PKI

In order to send a piece of data across the VICINITY system and keep it secure, there is a need to choose a reliable mechanism that provides an end-to-end encryption in P2P network environment. Usage of one pre-shared key for encryption and decryption using symmetrical encryption means that both sides need to know the same cryptographic key to communicate successfully, which is not viable for a P2P network of VICINITY end points. Asymmetrical cryptography mechanisms need to be put in place.

Out of many, public key cryptography stands out, as one of the strongest methods to deliver all security goals on all levels of technical area of defence in depth. The core pre-requisite when utilizing public key cryptography is to have two keys for every VICINITY end points. One public, available for any other endpoint that wishes to communicate, used to encrypt a message. The message is then decrypted by recipient's private key. This is used as a method of assuring the confidentiality, authenticity and non-reputability of data exchange among the P2P endpoints.

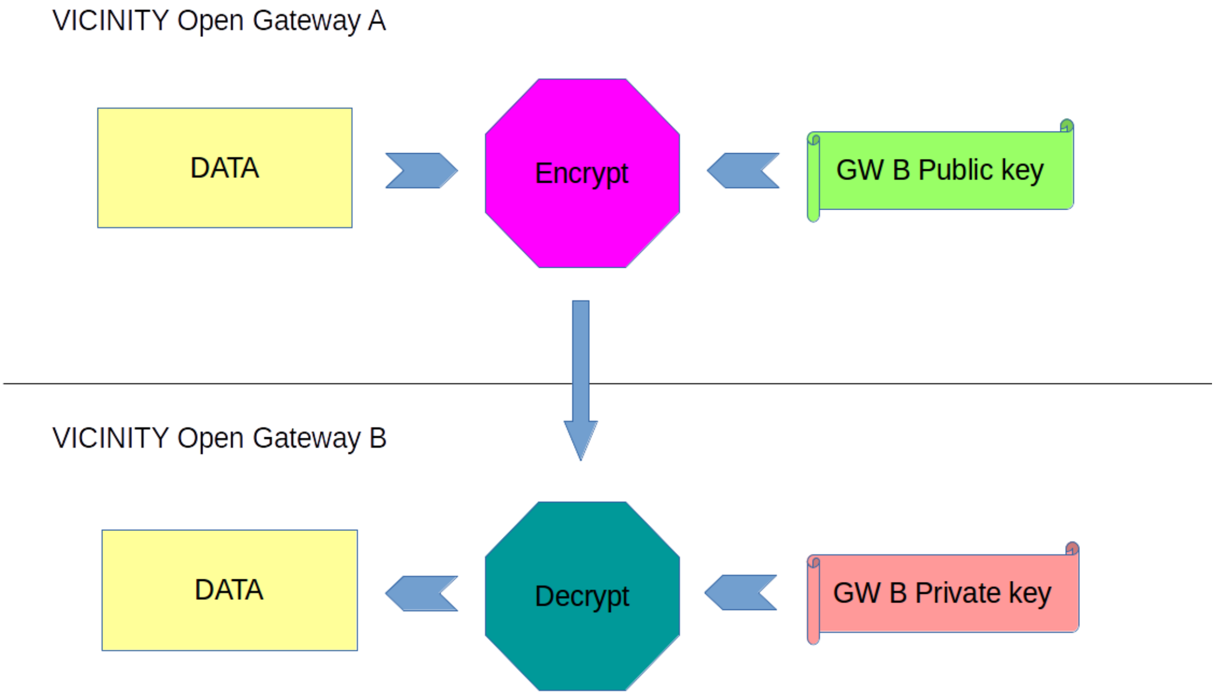

Figure 5 Public key cryptography, exemplar usage in VICINITY system.

There is however a significant disadvantage of using public key cryptography on small devices, and that are the high-performance requirements necessary to encrypt and decrypt reasonable amount of data in a short time. The public key algorithms known thus far are relatively computationally costly compared with most symmetric key algorithms of apparently equivalent security. The difference factor is the use of typically quite large keys and their ability to only encrypt relatively small blocks at one time. This has important implications for their practical use in VICINITY system.

One of the approaches that poses all advantages of public key cryptography while having reduced computational cost found in symmetrical encryption algorithms is a hybrid approach, that is commonly implemented not only in low-performance devices. A randomly generated secret key (a session key) is securely exchanged between the two P2P nodes, that wish to exchange data, using public key cryptography. This secret key is then used in symmetrical algorithm, in a schema known as authenticated encryption. This has the benefit of fast data encryption/decryption while still requiring a non-shared private key to get access to the key needed to decrypt the data.

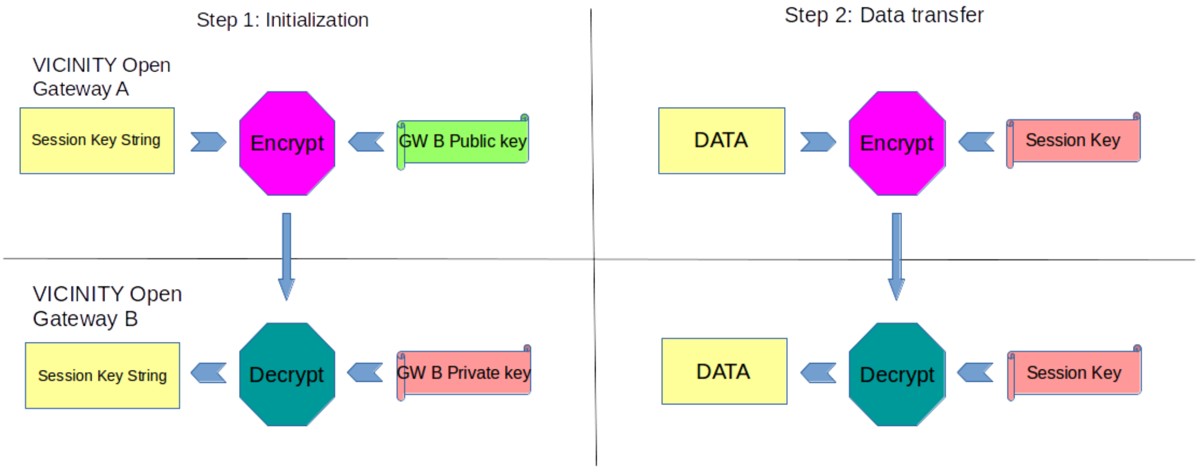

Figure 6 Authenticated encryption with session key exchanged using public key cryptography.

Now, when the secure data transfer is ensured, an issue of how to make sure that public keys are also exchanged securely, must be solved. For this, a set of mechanisms, services and approaches called public key infrastructure (PKI) has been developed and used over the past decades. The PKI is governed by a body known as the Public Key Cryptography Standards (PKCS). A PKI is an arrangement that binds public keys with respective identities of entities (like VICINITY gateways and organizations). The binding is established through a process of registration and issuance of certificates at and by a certificate authority (CA). [9]

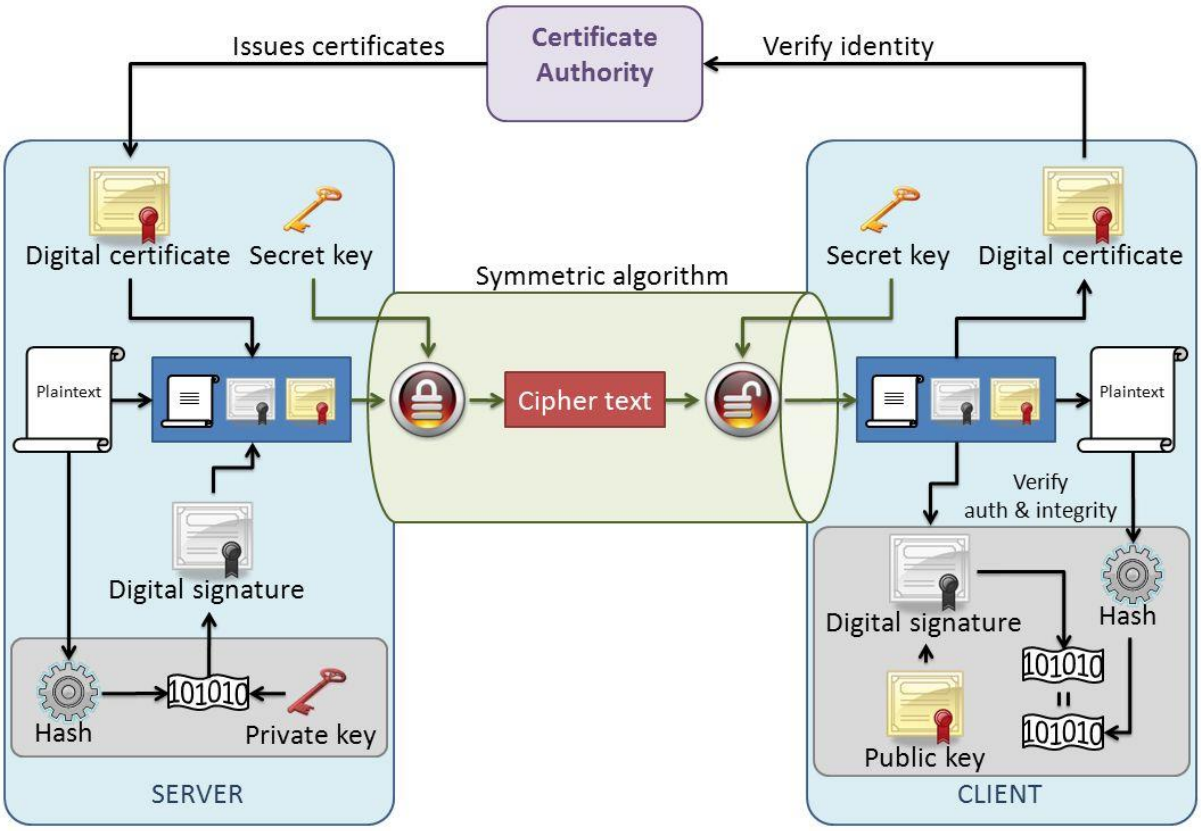

Figure 7 How the PKI actually works [10].

The PKI can come in various configurations, which will depend on the specific tools that will be decided upon over VICINITY system development. However, in general, this is how a PKI is structured and set up to run:

1. The request for a digital certificate is sent to the certificate authority.
2. After this specific request has been processed, the digital certificate is then issued to the entity that is requesting it.
3. The digital certificate then gets signed by confirming the actual identity of the end point who is requesting that specific digital certificate. The public key can now be used to further encrypt the plaintext (the session key) into the ciphertext that is sent from the sending VICINITY end point to the receiving VICINITY end point.

Effective use of PKI requires use of these certificates. However, effective use of certificates requires many additional services, such as:

- OCSP servers or CRL repositories,
- timestamping services,
- CA itself,
- etc.

As a consequence, client-side PKI tools must be implemented on VICINITY end points and they need to be able to discover and use these services. On the other hand, server-side PKI tools need to be able to provide these services and enable client tools to discover them. Concrete tools to be used during development is yet to be decided and it is likely that the decision will be changed over the course of implementation. This uncertainty is a consequence of proposed VICINITY authorization mechanisms, which will require some of the PKI components to cooperate closely with VICINITY Neighbourhood Manager.

Table 10 contains a list of tasks that have to be done in order for this security measure to function properly.

Table 10 List of tasks necessary for implementation of Authenticated encryption and PKI.

Authenticated encryption and PKI task list	**Affected VICINITY components**		
Task name	**Open Gateway API**	**Communication Server**	**Neighbourhood Manager**
Configure and deploy VICINITY CA services	No	Yes	No
Configure and deploy VICINITY RA services	No	Yes	No
Configure and deploy timestamping services	No	Yes	No
Configure and deploy OSCP services	No	Yes	No
Implement PKI client functionality for session key generation and exchange	Yes	No	No
Implement Authenticated encryption model for communication	Yes	No	No

4. VICINITY Security limitations

There are some limitations to the extent of which VICINITY security architecture can define measures that should be taken to guarantee security provided by the system. It is out of the scope of this deliverable to provide information about administrative part of defence in depth. There are no proposals for social engineering methods, no security training in any organization is described, etc. Also, it is out of scope of this document to define measures necessary to provide physical part of defence in depth (configuration of building alarm systems, etc.). Moreover, there are also limitations from the technical point of view, as noted in chapter 3.1 Achieving security goals.

Internet of Things privacy is the special considerations required to protect the information of individuals from exposure in the IoT environment, in which almost any physical or logical entity or object can be given a unique identifier and the ability to communicate autonomously over the Internet or a similar network. The data transmitted by a given endpoint might not cause any privacy issues on its own. However, when even fragmented data from multiple endpoints is gathered, collated and analysed, it can yield sensitive information. In VICINITY Neighbourhood manager, data owner has full control on which data by whom is controlled and processed.

Internet of Things security is also a special challenge because the IoT consists of so many Internet-enabled devices other than computers, which often go unpatched and are often configured with default or weak passwords [13]. Unless adequately protected, IoT things could be used as separate attack vectors or part of a thingbot. In a recent proof-of-concept exploit, for example, researchers demonstrated that a network could be compromised through a Wi-Fi-enabled light bulb. In December 2013, a researcher at Proofpoint, an enterprise security firm, discovered that hundreds of thousands of spam emails were being logged through a security gateway. Proofpoint traced the attacks to a botnet made up of 100,000 hacked appliances. As more and more products are developed with the capacity to be networked, it's important to routinely consider security in product development.

5. Conclusions

This deliverable discussed general goals of cybernetic security and technical measures that have to be applied and implemented into the VICINITY system, so the security of data transmitted across the P2P network can be considered guaranteed. Moreover, boundaries and limitations where this assurance is valid are described and justified.

The VICINITY Security architecture will be regularly evaluated and verified during Task 6.4 Security & Privacy evaluation of VICINITY Components particularly during:

- Integration of VICINITY Core components;
- VICINITY Lab testing;
- Deployment and evaluation in VICINITY Pilot sites.

Each security service implemented will be subject of security evaluation on the level of network, host or application. The level on which each security services will be evaluated is defined in the table 1. Identified finding needs to be evaluated from point of view of security risk and potential impact on business continuity. Findings will be summarized in a VICINITY security and privacy evaluation report - D6.4 including security issues justification. Changes of the security services resulted from the evaluation report will be implemented and summarized in D4.4 VICINITY client components continuous upgrades, first version and D4.5 VICINITY client components continuous upgrades, final version.

In general security services are putting “technical and process constraint” on environment where the VICINITY platform client components should run. It is possible to have specific issues in particular pilot sites setups, lab testing or open call projects implementation. It is expected to have minimum issues regarding the security services while the VICINIT Security architecture is defined with transparency in the mind. However, there might be specific changes of requirements in complex technology configurations. In case the such security issues will be risen the security risk and impact on business continuity will be evaluated and if necessary it will be addressed in deliverable D4.4 and D4.5.

List of Tables

List of Figures

References

[1] Andress, Jason. What is Information Security? pp. 1–16. doi:10.1016/b978-1-59749-653-7.00001-3 (https://doi.org/10.1016%2Fb978-1-59749-653-7.00001-3).

[2] ISO/IEC 27000:2016 (E). (2016). Information technology - Security techniques - Information security management systems - Overview and vocabulary. ISO/IEC.

[3] Aceituno, Vicente. "Open Information Security Maturity Model", http://www.ism3.com/node/39

[4] Rouse, Haughn, Gibilisco. Confidentiality, integrity, and availability (CIA triad). http://whatis.techtarget.com/definition/Confidentiality-integrity-and-availability-CIA

[5] Schneier on Security: Security in the Cloud. https://www.schneier.com/blog/archives/2006/02/security_in_the.html

[6] Defense in Depth: A practical strategy for achieving Information Assurance in today's highly networked environments. https://www.nsa.gov/ia/_files/support/defenseindepth.pdf

[7] Stewart, James Michael; Chapple, Mike; Gibson, Darril (2015). CISSP (ISC)2 Certified Information Systems Security Professional Official Study Guide.

[8] By Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=26067569

[9] What is a Public Key Infrastructure - A Simple Overview , April 17, 2015". http://www.net-security-training.co.uk/what-is-a-public-key-infrastructure/

[10]Darizotas. Understanding Secure Web communications: SSL/TLS and PKI. https://darizotas.blogspot.com/2013/02/understanding-secure-web-communications.html

[11]Fraser, B. (1997), RFC 2196 – Site Security Handbook, IETF

[12]Macauley, Hamilton, Rabeler. "Transparent Data Encryption (TDE), October 25, 2015". https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption

[13]Debruce. Proofpoint Uncovers Internet of Things (IoT) Cyberattack. https://www.proofpoint.com/us/proofpoint-uncovers-internet-things-iot-cyberattack

[14]http://www.vicinity-h2020.eu

[15]ICT 30 – 2015: Internet of Things and Platforms for Connected Smart Objects - http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/914-ict-30-2015.html