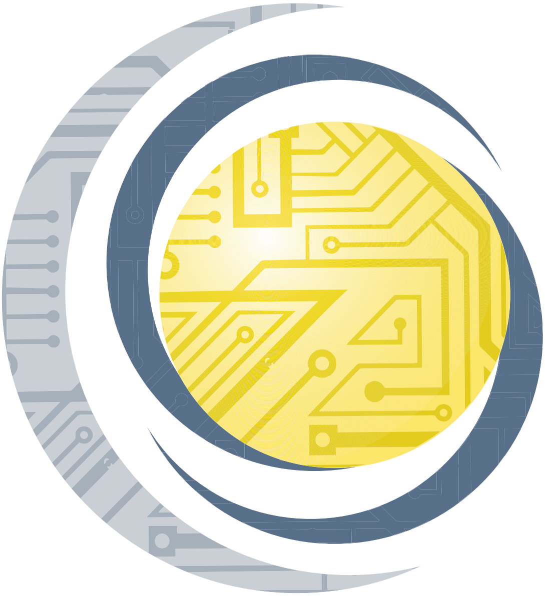
Revised Payments Service Directive: A Blockchain-based Implementation Model

Printed in February 2018



Revised Payments Service Directive:

A Blockchain-based Implementation Model

GLOBAL
CYBER SECURITY
CENTER

CONTENTS

Revised Payments Service Directive:

A Blockchain-based Implementation Model

Corona F., Faramondi L., Leccese F., Peverini F.

13th February 2018

Abstract

The definition of the Payments Service Directive 2 (PSD2) represents an opportunity to design a novel model about the relations among clients, banks, and third party providers. The aim of this document is to provide an overview about the applicable technologies useful to realize a new implementation model of the directive.

1- INTRODUCTION

EU Member states have until 13th January 2018 to transpose the directive 2015/2366/UE - also known as PSD2 - into their national legislation [1]. The directive not only designs new strategies to protect the consumer and to develop new payment solutions, but also focuses on the introduction of three new players in the market. Each player is part of the new model and provides a fundamental service about the payment system. On the one hand, the introduction of the PSD2 represents a business issue because of the loss of the direct link between the bank and the customer; on the other hand, it may represent an opportunity to redesign the chain of the services involved in the payments [2]. Indeed, it may be possible to introduce new technologies by importing expertise and knowledge from different sectors, with the aim of finding solutions to the challenges introduced by the new legislation.

The proposal of redesigning the framework discuss in this work focuses on the introduction of three macro innovations. The first proposal consists in the introduction of the distributed ledger technology as archiving tool of banking transactions. BitCoin, Ripple and Ethereum are three examples of projects related to the use of the distributed ledger in the economic and financial fields. In comparison with the current storage system of the banking transaction the preservation of the integrity of data is the main advantage introduced by the new system.

Another hint for the framework redesign, also supported by [3] and [4], goes along with the adoption of the standard open banking. Indeed, the development of open API for the use of bank services goes perfectly with the new actors, as costumers' services providers, introduced by the PSD2.

Another innovation introduced in this work is the use of cryptographic techniques - indeed they are taking root in the field of cloud computing. They appear to be fundamental since, as required by the PSD2, services managed by third parties require the partial access to documents and data of other agents in order to analyze them without having the full access.

2. PRELIMINARIES

For the sake of clarity in this section we have collected some necessary preliminaries about the adopted technologies, encryption methods, and digital signs.

2.1 Cryptography Overview

The fundamental objective of cryptography is to enable two people, Alice and Bob, to communicate over an insecure channel in such a way that an opponent, Oscar, cannot understand what is being said. This channel could be a telephone line or a computer network, for example. The information that Alice wants to send to Bob is called plaintext and can be a mail or numerical data for example. In this chapter we provide a brief introduction to classical and modern cryptography, with special attention to the hash functions.

In this section we analyse the concepts used in classical cryptography, that is a scenario where Alice and Bob use the same key to encrypt and decrypt the information. A cryptosystem is a suite of cryptographic algorithms used over the process of encryption of a data, formally:

Definition 1. A cryptosystem is a five-tuple (**P**; **C**; **K**; **E**; **D**), with the conditions:

1. **P** is a finite set of all possible plaintexts;

2. **C** is a finite set of all possible ciphertexts;

3. **K** is a finite set of all possible keys, called keyspace;

4. For each k **K** there is an encryption rule $e_k \in$ **E** and a corresponding decryption rule $d_k \in$ **D**. Each ek : **P** $\longrightarrow$ **C** and dk : **C** $\longrightarrow$ **D** are functions such that

$$d_k\,(e_k\,(x)) = x$$

for every plaintext element $x \in$ **P**.

Property 4 is fundamental, it says that if a plaintext x is encrypted using ek, and the resulting ciphertext y is subsequently decrypted with dk , then we obtain x. Alice and Bob will employ the following protocol to use a specific cryptosystem: first, they choose a key k. To securely exchange this key, we just suppose Alice and Bob have chosen k in a secret room, not being observed by anyone else. At a later time, Alice wants to communicate to Bob the message $x = x_1\ x_2 \ldots x_n$, a string of length n≥1, where all $\in x_i$ **P**. Each x_i is encrypted using e_k , hence Alice computes $y_i = e_k\ (x_i)$ and sends the resulting ciphertext $y = y_1\ y_2$ … yn to Bob over the channel. Bob, once received y, decrypts it using d_k , obtaining x.

We need to make a clarification about ek : it can not happen that

$$y= e_k\ (x_i) = e_k\ (x_j)$$

for i ≠ j, otherwise d_k could not be accomplished in an unambiguous manner. To avoid this situation it is sufficient to choose ek such that it is an injective function.

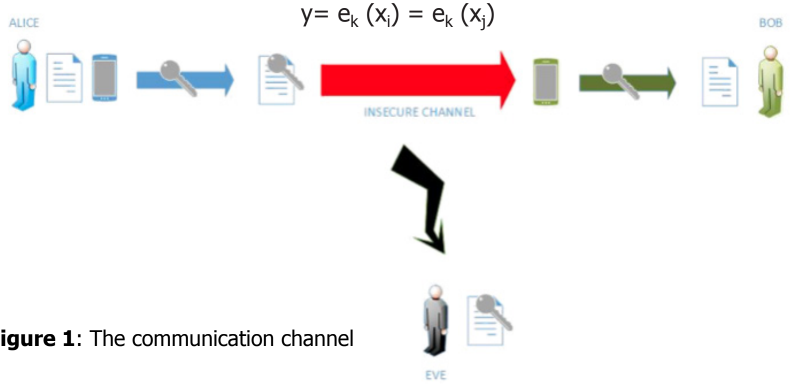


Figure 1: The communication channel

We make a classic example of a cryptosystem, the Substitution Cipher. In order to encrypt an English text with this cipher we must set up a correspondence between alphabetic characters and number from 0 to 25 in the spontaneous way:

$$A \longleftrightarrow 0;\ B \longleftrightarrow 1;\ \ldots\ ;\ Z \longleftrightarrow 25.$$

Cryptosystem 1. Substitution Cipher
Let **P** = **C** = {0; 1; … ; 25}, K consists of all possible permutation of the alphabet, for example

a	b	c	d	e	f	g	h	i	j	k	l	m
X	N	Y	A	H	P	O	G	Z	Q	W	B	T

n	o	p	q	r	s	t	u	v	w	x	y	z
S	F	L	R	C	V	M	U	E	K	J	D	I

and for each п ∈ **K**, define $e_п(x_i) = п(x_i)$ and $d_п(y_i) = п^{-1}(y_i)$.

Thus, e_k (a) = X, e_k (i) = Z and so on. The decryption function works in the same way but with the inverse table:

A	B	C	D	E	F	G	H	I	J	K	L	M
d	l	r	y	v	o	h	e	z	x	w	p	t

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	g	f	j	q	n	m	u	s	k	a	c	i

Hence $d_{п}(A) = d$, $d_{п}(G) = h$ etc.

Once Alice and Bob agree they can start the communication, for example suppose that Alice wants to send to Bob the message

x = revisedpaymentservicedirective;

after the encryption she obtains:

$$\begin{array}{l*{30}{c}}
\mathbf{x} = & r & e & v & i & s & e & d & p & a & y & m & e & n & t & s & e & r & v & i & c & e & d & i & r & e & c & t & i & v & e \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\mathbf{y} = & C & H & E & Z & V & H & A & L & X & D & T & H & S & M & V & H & C & E & Z & Y & H & A & Z & C & H & Y & M & Z & E & H
\end{array}$$

and sends it to Bob. Bob has just to use the inverse table (easily obtainable by п) to decrypt y and finally read Alice's message:

$$\begin{array}{l*{30}{c}}
\mathbf{y} = & C & H & E & Z & V & H & A & L & X & D & T & H & S & M & V & H & C & E & Z & Y & H & A & Z & C & H & Y & M & Z & E & H \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\mathbf{x} = & r & e & v & i & s & e & d & p & a & y & m & e & n & t & s & e & r & v & i & c & e & d & i & r & e & c & t & i & v & e
\end{array}$$

Substitution Cipher is a simple example of Classical Cryptography that shows how the encryption-decryption process takes place. Modern cryptosystems are based on more complex algorithms and take advantage of more and more sophisticated mathematical tools but they still follow Kerckhoffs' principle, z list stated by Dutch cryptographer Auguste Kerckhoffs in the 19th century:

1. The system must be practically, if not mathematically, indecipherable;

2. It should not require secrecy, and it should not be a problem if it falls into enemy hands;

3. It must be possible to communicate and remember the key without using written notes, and correspondents must be able to change or modify it at will;

4. It must be applicable to telegraph communications;

5. It must be portable, and should not require several persons to handle or operate;

6. Lastly, given the circumstances in which it is to be used, the system must be easy to use and should not be stressful to use or require its users to know and comply with a long list of rules.

This list can be summarized in: "A cryptosystem should be secure even if everything about the system, except the key, is of public knowledge".
In accordance with Kerckhoffs' principle, the models proposed by us are exposed without any secrecy about encryption algorithms involved or related details.

Hash Function

A cryptographic hash function is a mathematic algorithm that can provide assurance of data integrity. A hash function is used to construct a short digest (usually 160 or 256 bit) of some data; if the data is changed, then the digest, or fingerprint, will be no longer valid. Even if the data is stored in an unsafe place, for example the net, its integrity can be checked from time to time by recomputing the fingerprint and verifying if the digest is different from the original. This algorithm uses some special mathematic functions called "one way", i.e. a function that can't be inverted easily. It allows using hash function in many aspects of cyber security as authentication, password protection and mostly digital signature. A hash function has three important properties:

1. It must be easy to calculate;

2. It has not to be invertible or, if it is possible to invert, it's necessary too much time;

3. It has to be almost impossible to find the same fingerprint twice for different data.

Let h be a hash function and let x be some input data, for example a binary string of arbitrary length, then the corresponding fingerprint is defined to be $y = h(x)$. Moreover, also families of keyed hash functions exist, i.e. if we suppose that Alice and Bob share a secret key, k, which determines a hash function, say h_k , then the corresponding fingerprint $y = h_k(x)$ can be calculated just by Alice and Bob. Notice the distinction between the assurance of data integrity provided by an unkeyed hash function, as opposed to a keyed one. In the first case, the message digest must be secretly stored so it cannot

be altered; on the other hand, if they use a keyed hash function then they can transmit both the data and the digest over an insecure channel.
Suppose now that h is an unkeyed hash function, x is some input data and define the fingerprint $y = h(x)$. A hash function is considered secure if it's impossible to solve the following three problems:

1. Preimage. Given h and y, find x such that $y = h(x)$;

2. Second Preimage. Given h and x, find x′ such that $h(x) = h(x')$;

3. Collision. Given h, find x; x′ such that $x \neq x'$ and $h(x) = h(x')$.

The most common hash functions are part of the iterated hash functions. The iterated hash functions are a particular technique by which a compression function, say compress, can be extended to an hash function with an infinite domain. Let now see an example of iterated hash function:

1. Preprocessing step. Given a bitstrings x, construct a string, using a public algorithm (padding function)

$$x = x_1 \,||\, x_2 \,||...||\, x_n:$$

2. Processing step. Let IV be a public initial value, which is a bit string. Then compute:

$$IV = y0,$$

$$\text{compress}(y_0 \,||\, x_1) = y_1$$

$$\vdots$$

$$\text{compress}(y_{n-1} \,||\, x_n) = y_n$$

The last step will be the output of the iterated function, so

$$y_n = y = h(x):$$

The pre-processing step must guarantee that the function isn't injective to make the problems seen before impossible. Most hash functions used in practice are in fact iterated hash functions and can be viewed as special cases of the generic construction described above, as for example all the Secure Hash Algorithms (SHA).

Digital Signature

As said above, one of the possible applications of a hash function is the Digital Signature. A signature is used in everyday situations such as writing a letter, withdrawing some money from a bank, signing a contract, etc. Similarly a Digital Signature is a method to sign a message stored in an electronic form that can be transmitted over a computer network. It guarantees that the signed digital document was generated by the singer, has not been tampered with and that the signature cannot be rejected. With a conventional signature, the sign is a part of the document, however the digital signature is not phisically attached to the document, so the algorithm used must bind the signature to the message. The verification of the Digital Signature is far different from the verification of the other signatures, because it's necessary to use a publicly known verification algorithm thus everyone can verify that. So the use of a secure Digital Signature will prevent the possibility of forgeries. Finally a last big difference between digital and conventional signature is the fact that the copy of the signed message is identical to the original. This feature means that care must be taken to prevent a signed digital message form being reused, for example the message must contain some information such as date or timestamp. A digital signature scheme consists in two algorithms: sig_k a signature algorithm and ver_k a verification algorithm. Both of them should be polynomial time functions. Moreover, given a message x, it should be computationally infeasible for anyone other than Alice to calculate a valid signature y. The most used digital signature algorithms are DSA and ECDSA.

2.2 The Blockchain Technology

Inter-bank payments are usually performed using a central counterpart and every bank has a local database, that acts as authoritative ledger where all account balances and transactions are recorded. From a technical point of view, the Blockchain is defined as a distributed replicated database that allows secure transactions without a central authority. The first Blockchain implementation is the cryptocurrency Bitcoin. In this context, the replicated database acts as global ledger tracking all cryptocurrency transactions between participants.

Blockchains allow different parties that do not trust each other to share information without requiring a central administrator. Transactions are processed by a network of users acting as a consensus mechanism so that everyone is creating the same shared system of record simultaneously. The value of decentralized control is that it eliminates the risks of centralized control. With a centralized database, anybody with sufficient access to that system can destroy or corrupt the data within. This makes users dependent on the administrators.

The Blockchain is composed by the concatenation of fundamental parts also known as blocks. A block is a data structure composed by the following fields:

- Block Number: An integer unique number that represents the block;

- Nonce: An integer random number;

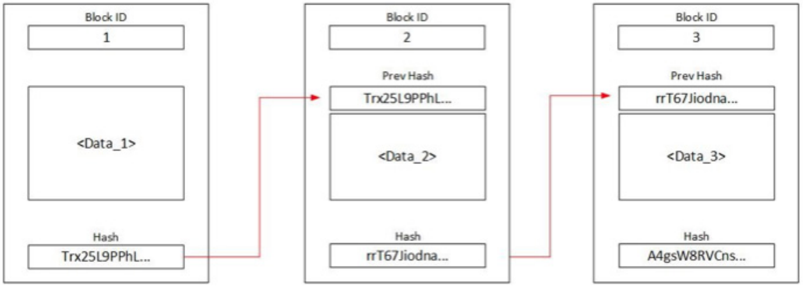
- Data: The set of information stored in the Blockchain (e.g. transactions, contracts, documents, etc...);

- Hash: the hash code of the block. It depends on all the previous fields. More over this alphanumerical string is characterized by the presence of four initial characters equal to 0. See Section 2.1 for more details.

Respect to the canonical databases, the Blockchain, due to the block structure, provides an intrinsic information integrity verification. A block is said to be valid if it respects all the predeterminated features, for example in Bitcoin's Blockchain a block is valid if and only if its hash solve a particular cryptographic puzzle and so the hash starts with 4 zeroes. This validation process for a given block, characterized by its unique number and its set of stored information, consists in the research of a random number (nonce) such that the hash code respects the rule about the initial zeros. This validation process is also known as Proof-of-Work. The PoW is mandatory because the parts involved don't trust each other.
Thanks to the properties of the hash functions, even the slightest change in any field of the block produces a new hash code that does not respect the integrity scheme.The Blockchain is defined as a list of blocks where each block is linked to the previous block. In more details, in addition to the previously defined set of fields, each block contains also the reference to the previous block in the chain as shown in Fig.4.
The task of validating the new blocks is demanded to a peer-to-peer net-works of miners. The peer-to-peer network guarantees transactional security throughout the consensus process. Every time a new block is validated, each node verifies the block and updates its local copy of the database, adding a new block to the chain. Nodes follow the protocol that is embedded in the Blockchain software, which determines a single state of the database even if there is no single authoritative copy of it. While in a centralized architecture there is a single authoritative database operated by a single entity, in the Blockchain every node has a local synchronized copy, due to this end each Blockchain copy of each miner must be aligned to the other copies in the network.

Due to the presence of a lot of synchronized Blockchain copies in the peer-to-peer network, it is pretty hard to modify the data in a Blockchain without creating an alignment problem in the network.

Figure 2: Blockchain structure



Suppose to change the data in a single copy of the Blockchain in the network. Suppose, moreover, to change the data not collected in the most recent block. Due to the structure of the block, the change has a ripple effect and it affects all the other blocks. This is due to the automatic violation of the new hash code in the compromized block. As introduced, the hash code depends on all the fields of the blocks and every modification implies a new different hash code of the block. The ripple effect is due to the presence of the reference to the previous block in each item of the chain.

Suppose now, the presence of a malicious cunning miner. He/she is interested in modifying the data as just described. In this case, with the aim to obtain valid blocks, he/she applies the following strategy:

1. The attacker changes the information stored in an old block;

2. With the aim to validate again the modified block, he/she applies the mining activity and finds a new nonce such that the associated hash has four initial zeros;

3. Due to the new hash, all the following blocks are now not verified. The attacker iteratively validates all the following blocks by finding new nonce values and new hash codes, until the last block of the chain.

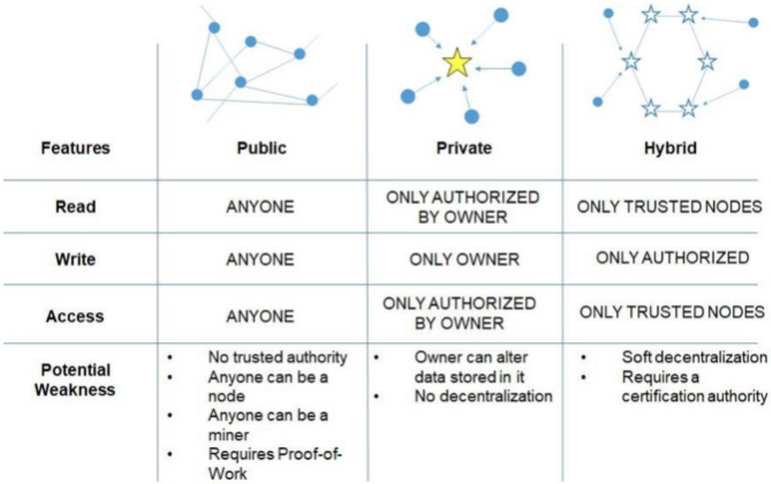
Apparently, now the attacker is succeeded in ensuring a valid Blockchain but the hash code of the last block is different to the last hash of the other Blockchain copies in the network. The misalignment about the last hash code produces an alert.

Based on network access permission, there are three categories of Blockchain:

Public: network access is free and anyone can set up a node to validate transactions;

Private: network access is restricted to a set of known participants, au-thorized by the Blockchain’s owner;

Hybrid: network access is restricted to a set of known participants, that are recognized as trusted nodes.



Features	Public	Private	Hybrid
Read	ANYONE	ONLY AUTHORIZED BY OWNER	ONLY TRUSTED NODES
Write	ANYONE	ONLY OWNER	ONLY AUTHORIZED
Access	ANYONE	ONLY AUTHORIZED BY OWNER	ONLY TRUSTED NODES
Potential Weakness	• No trusted authority • Anyone can be a node • Anyone can be a miner • Requires Proof-of-Work	• Owner can alter data stored in it • No decentralization	• Soft decentralization • Requires a certification authority

Figure 3: Summuary of typers of Blockchain

In permission-less systems, a reward in cryptocurrency is usually given to nodes for each validated block as an incentive to join and protect the network. While this was typical for cryptocurrency permission-less systems, in new generation permissioned or hybrid Blockchains an incentive mechanism is not required. This architecture is more suited for enterprise solutions where the authentication of both nodes and participants is usually required.

2.3 Homomorphic Encryption

One of the biggest problem with public-key cryptography is the keys management, in some circumstances the definition of a keys register and its access level can cause frictions or legal disputes. Otherwise the manipulation of big data, especially if encrypted, can require a huge quantity of time and computation to be done, depending on which cryptosystem is used. A possible solution for these two issues is the Homomorphic Encryption, a particular cryptographic technique that allows evaluating circuits over encrypted data without knowing the decryption key. To better understand this topic it is useful to make a simple example: imagine that Alice and Bob choose secretly a number, for example Alice chooses 4 and Bob 6, they can evaluate the sum of their numbers, 10, and communicate this result without reveal their secret numbers. Homomorphic Encryption works in the same way, making this possible to obtain plaintext data by manipulating encrypted without any decryption.

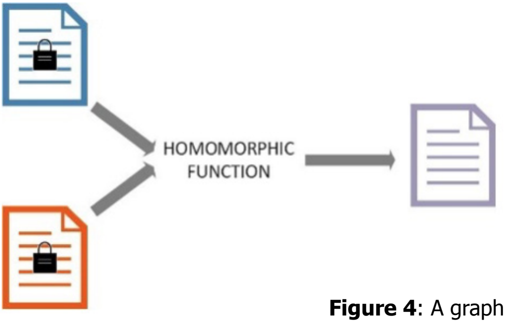


Figure 4: A graphic presentation

A more useful example of homomorphic encryption is given by RSA cryptosystem. In the RSA the public key is the modulus m and the exponent e, the encryption function is the exponentiation of a message x, the numerical representation of a text message, to the exponent e:

$$(x) = x^e \bmod m:$$

It is enough to remember the properties of exponentation, specifically that $a^m b^m = (ab)^m$, to notice that holds:

$$e(x_1)e(x_2) = x_1^e x_2^e \bmod m = (x_1 x_2)^e \bmod m = e(x_1 x_2):$$

Table 1: Final considerations over Homomorphic Encryption

Pros	Cons
Privacy Easy recall process Most advanced cryptography Keys remain private	Only theoretical Very slow Requires scientific research

So in the RSA, starting with two different encrypted messages, one can easily calculate their product mod m to get a correctly encrypted new message without knowing the private exponent e. This is not a weakness of the RSA, since the product of two random messages almost never gives a sensible result, but shows the huge potential of homomorphic functions.

Unfortunately this is one of the very few examples of applied homomorphic encryption, new research is needed to make this technology applicable to the digital payments scenario.

2.4 Open Banking Preliminaries

In this section, the two most relevant elements of the open banking framework are described and analysed with the aim to propose an improvement about the implementation of the PSD2 in terms of information sharing.

2.4.1 Web Services

A Web Service consists in a software architecture able to allow the interoper-ability among multiple agents in a distributed environment. Each agent in the network presents a list of deliver services useful to the other agents to com-plete a task. Each request of service consists in a request of a remote operation executed adopting several protocols (e.g. SOAP, XML, and HTTP).
One of the most interesting characteristics of this software architecture consists in the adoption of several implementation languages. Each agents is able to interact with the other agents by requesting the services without speaking a common language thanks to the presence of a common layer represented by the interface of the services.
Some of the advantages about the adoption of Web Services:

- Allows interoperability among different software and hardware platforms. Adopts open protocol and standards;

- Thanks to the use of HTTP protocol, the web services inherit all the characteristics about the message security;

- The interaction of several services provides achievements of complex services.

The protocol stack of the web service architecture consists in a set of network protocols used to define, realize, and allow the interactions among multiple services. This stack is composed by four areas:

- Transport Service : it is responsible of the message delivery among the net-work applications. It includes the HTTP, SMTP,FTP, XMPP protocols;

- XML Messaging : This standard defines the structure of the messages. The adoption of XML tags is provided in several protocols (e.g. SOAP, REST, and JAX-RPC);

- Service Description : It consists in a public software interface implemented in WSDL common language. It is an XML-based interface and comprehends the details about the services and their use;

- Services List : It represents a centralization of several information about the available web services in the networks thanks to the UDDI (Universal Description Discovery and Integration).

2.4.2 Open API

An open API consists in a public application programming interface able to provide to developers a set of useful tools for the access to a proprietary software application or a web service. The APIs are a set of commands that dene how one application can communicate and interact with another. In the context of this work we are interested to the APIs that allow one piece of software to interact with a remote server over an external TCP/IP-based network. Open APIs have three main characteristics:

- Usually, Open APIs use is not restricted in some cases the access is regulated by a registration to the service;

- An Open API may be free to use but the publisher may limit how the API data can be used;

- Open API is based on an open standard.

Often the use of Open APIs is publicly available for all the allowed or registered users. The APIs owner allows the use of its services and data. However, it is important to remember that opening back end information to the public can create a range of security and management challenges. If in one hand the sharing of services and data can improve the business incomes, in the other hand publishing open APIs can make it harder for organisations to control the experience end users have with their information assets. Open API publishers cannot assume client apps built on their APIs will offer a good user experience. Furthermore, they cannot fully ensure that client apps maintain the look and feel of their corporate branding.

One of the effects of the increasing use of open APIs is the increasing number of social platform over the web. Some open APIs retrieve data from databases behind a website and these are called Web APIs. For example, Google's You-Tube API allows developers to integrate YouTube into their applications by providing the capability to search for videos, retrieve standard feeds, and see related content.

Web APIs are used for exchanging information between a local software application and a remote website either by receiving or by sending data. When a web API fetches data from a website, it opens a new data exchange session and the application makes a carefully constructed HTTP request to the server the site is stored on. In case of a data request message, the server then sends data back to the application using the expected data format, otherwise, in case of a data modification request, the remote service incorporates the data changes to the website database.

An example of useful open APIs applicable in the context of the PSD2 implementation is presented by Capital One Financial Corporation a bank holding company specialized in credit cards, auto loans, banking and saving products . Among the products presented by Capital One we have selected three sets of open APIs that lend themselves to the implementation of the PSD2 services:

- Bank Account Starter APIs allows the customers to submit an application for a new banking account. Customers can apply for an individual account or a joint account (up to two applicants);

- Credit Offers APIs returns a personalized list of credit card offers in under 60 seconds based on just a few pieces of personal information;

- Rewards APIs allows banks to get information on the miles, points or cash rewards their customers have earned with their accounts. Based on their individual rewards balances and redemption opportunities the rewards plans are completely customizable.

3. REVISED PAYMENTS SERVICE DIRECTIVE

The European Payment Services Directive, better known as PSD, was published in 2007 and defines a modern and coherent Community legal framework for electronic payment services. The PSD sets several goals:

1. to regulate the access to the market to foster competition in the provision of services;

2. to ensure greater transparency and user protection;

3. to standardize rights and obligations in the provision and use of payment services to set on the legal bases for the realization of the SEPA;

4. to encourage the use of electronic payment instruments to reduce the cost of inefficient tools such as paper and cash;

5. to stimulate the competition and participation in the European Union in the payments industry, including the non-banks' involvement.

The innovative PSD was transposed into national law between 2009 and 2010. In the wake of this Directive in July 2013 the European Union established a commission devoted to the revision of the PSD and after two years the final text of the PSD2 was published in the Official Journal of the EU on December. PSD2 transposition within 20 days was compulsory, while the entry into force of the new Directive was set in January 2018.

3.1 Aims of PSD2

The new directive has become necessary for a number of reasons: first and foremost, the need to regulate an increasingly fragmented market characterized by increasing complexity in terms of players and digital evolution; on the other hand the desire to harmonize the European regulatory framework as not all States have adopted the PSD in the same way.
The PSD2 is only the latest EU legislative action in the field of payment services to create an integrated single market by aligning the rules between Payment Service Providers (PSPs) and new market players nowadays not regulated (AISPs, PISPs), to strengthen system security and to ensure greater competition for customers.

3.2 Context

Technological and information technology has completely overturned the payment market, which has added the opening of the European Union to international markets and the introduction of new players and service providers. According to world data, non-electronic transactions are steadily increasing, +9% in 2014 compared to the previous year.

Furthermore, the growth in digitization of payments is not going to fall and next year is expected to be 19% higher than last year, thanks also to this new Directive.

In our Country cash is still very much used in comparison to the rest of the world, about 86% of payments are non-digital, according to ISTAT data, but digital transactions are already having a positive trend since 2013-2015 (+10%) thanks to increased online purchases and subsequent use of cards (+91% of on-line payment over the last 5 years). The causes of increasing digitization are:

1. the increase in the use of technology tools, such as tablets or smartphones, by customers;

2. a change in the habits of consumers who find these payment methods more efficient and with one best customer experience;

3. new marketing strategies that tend to focus on the use of electronic payment instruments to collect information related to customer behaviour.

Finally, the entry of new players is increasing the level of competition, the over the top (Google, Amazon, Facebook, Apple...) and the new TPPs (Sofort, Trustly, ...) are changing the traditional context of the bank service providers creating new business models with which banks must compare. Principally in other EU Countries the new competitors are gaining market shares and the risk of disintermediation in customer relationship with Banks and other traditional traders is increasingly present.

In this context, the new PSD2 Directive is concerned with stimulating the use of innovative digital tools, regulating security standards and at the same time regulating existing services and payment practices.

3.3 Central Authority

With the new PSD2 Directive, the European authorities intend to promote greater interaction and cooperation between the various authorities in the Member States.For this purpose during the last year, two important documents were published: “Passporting Rules” and “EBA Register” with a view to define rules for co-operation and information exchange between the authorities of Member States and the technical requirements for the development and management of the central electronic register, governed by article 14,15,16:

Article 14
Registration in the home Member State

1. Member States shall establish a public register in which the following are entered:

(a) authorised payment institutions and their agents;

(b) natural and legal persons beneting from an exemption (as for example in 3.5.2);

(c) the institutions that are entitled under national law to provide payment services.

Branches of payment institutions shall be entered in the register of the home Member State if those branches provide services in a Member State other than their home Member State.

2. The public register shall identify the payment services for which the payment institution is authorised or for which the natural or legal person has been registered. Authorised payment institutions shall be listed in the register separately from natural and legal persons beneting from an exemption. The register shall be publicly available for consultation, accessible online, and updated without delay.

3. Competent authorities shall enter in the public register any withdrawal of authorisation and any withdrawal of an exemption.

4. Competent authorities shall notify EBA of the reasons for the withdrawal of any authorisation and of any exemption.

Article 15
EBA register

1. EBA shall develop, operate and maintain an electronic, central register that contains the information as notified by the competent authorities in accordance with paragraph 2. EBA shall be responsible for the accurate presentation of that information.
EBA shall make the register publicly available on its website, and shall allow for easy access to and easy search for the information listed, free of charge.

2. Competent authorities shall, without delay, notify EBA of the information entered in their public registers in a language customary in the field of finance.

3. Competent authorities shall be responsible for the accuracy of the information and for keeping that information up-to-date.

4. EBA shall develop draft regulatory technical standards setting technical requirements on development, operation and maintenance of the electronic central register and on access to the information contained therein. The technical requirements shall ensure that modification of the information is only possible by the competent authority and EBA.

5. EBA shall develop draft implementing technical standards on the details and structure of the information to be notified, including the common format and model in which this information is to be provided.

Article 16
Maintenance of authorisation

Where any change affects the accuracy of the information and the evidence provided, the payment institution shall, without undue delay, inform the competent authorities of its home Member State accordingly.
The establishment of this register aims to strengthen the transparency of the operation of authorized payment institutions, held at the European Banking Authority "EBA", which will include all the information regarding the payment institutions registered with the individual national banks.
Furthermore the content of the EBA Register will be available on its website whereas, in order to keep it up-to-date, National Authorities will have to notify the EBA without delay of the information entered in their respective National Registers, the accuracy of which will remain in any case responsible.

3.4 The new Third Part Providers

The entry into force of PSD2 introduces for users who use an online account the possibility of making payments or accessing bank reporting through authorized third-party software (PISP, AISP).

3.4.1 AISP

The AISP, Account Information Services Provider, allows the customer to aggregate information from multiple bank accounts.
This is the main innovation of the PSD2 because currently doesn't exist any

service provider that allows to do so. Obviously a customer can choose one or more AISP depending on his needs and on the services offered and these can only act after a specific customer consent. So AISPs can connect to accounts and retrieve information to provide the customer with an overview of its financial situation, an analysis of spending habits in an easy and interactive way. Furthermore these services will be useful to monitor investments and support financial planning. The customer's authentication to the AISP has to be regulated by the SCA, instead the banks must provide APIs to authorize AISP to access bank accounts and information.
The AISP should provide services only where based on the customer's explicit consent.

Article 64
Consent and withdrawal of consent

1. Member States shall ensure that a payment transaction is considered to be authorised only if the payer has given consent to execute the payment transaction. A payment transaction may be authorised by the payer prior to or, if agreed between the payer and the payment service provider, after the execution of the payment transaction.

2. Consent to execute a payment transaction or a series of payment transactions shall be given in the form agreed between the payer and the payment service provider. Consent to execute a payment transaction may also be given via the payee or the payment initiation service provider. In the absence of consent, a payment transaction shall be considered to be unauthorised.

3. Consent may be withdrawn by the payer at any time, but no later than at the moment of irrevocability in accordance with Article 80. Consent to execute a series of payment transactions may also be withdrawn, in which case any future payment transaction shall be considered to be unauthorised.

4. The procedure for giving consent shall be agreed between the payer and the relevant payment service provider(s).

Article 67
Rules on access to and use of payment account information in the case of account information services

1. Member States shall ensure that a payment service user has the right to make use of services enabling access to account information as referred to in point (8) of Annex I. That right shall not apply where the payment account is not accessible online.

2. The account information service provider shall:

(a) provide services only where based on the payment service user's explicit consent;

(b) ensure that the personalised security credentials of the payment service user are not, with the exception of the user and the issuer of the personalised security credentials, accessible to other parties, and that when they are transmitted by the account information service provider, this is done through safe and efficient channels;

(c) for each communication session, identify itself towards the account servicing payment service provider(s) of the payment service user and securely communicate with the account servicing payment service provider(s) and the payment service user;

(d) access only the information from designated payment accounts and associated payment transactions;

(e) not request sensitive payment data linked to the payment accounts;

(f) not use, access or store any data for purposes other than for performing the account information service explicitly requested by the payment service user, in accordance with data protection rules.

3. In relation to payment accounts, the account servicing payment service provider shall:

(a) communicate securely with the account information service providers;

(b) treat data requests transmitted through the services of an account in-formation service provider without any discrimination for other than objective reasons.

4. The provision of account information services shall not be dependent on the existence of a contractual relationship between the account information service providers and the account servicing payment service providers for that purpose.

3.4.2 PISP

After the PSD2 you can avoid entering a credit card at a purchase, but you will be able to authorize a business to handle the start-up phase of the payment, the PISP (Payment Initiation Services Provider), by charging the expense directly to the bank account.
Therefore the PISP is a service provider which acts as an intermediary between the bank and the customer providing the initiation to the payment.
It is important to point out that in all this process it must be forbidden to the PISP to access the customer's account, but it should be limited to act as an intermediary.
Also in this case the customer's authentication must be regulated by the SCA. The customer has to give explicit consent payment to be executed, as established in the Directive (art. 64 and art. 66). Furthermore the customer can't in anyway revoke the payment order after giving consent to the PISP (art. 80).

Article 66

Rules on access to payment account in the case of payment initiation services

1. Member States shall ensure that a payer has the right to make use of a payment initiation service provider to obtain payment services as referred to in point (7) of Annex I. The right to make use of a payment initiation service provider shall not apply where the payment account is not accessible online.

2. When the payer gives its explicit consent for a payment to be executed in accordance with Article 64, the account servicing payment service provider shall perform the actions specified in paragraph 4 of this Article in order to ensure the payer's right to use the payment initiation service.

3. The payment initiation service provider shall:

(a) not hold at any time the payer's funds in connection with the provision of the payment initiation service;

(b) ensure that the personalised security credentials of the payment service user are not, with the exception of the user and the issuer of the personalised security credentials, accessible to other parties and that they are transmitted by the payment initiation service provider through safe and efficient channels;

(c) ensure that any other information about the payment service user, obtained when providing payment initiation services, is only provided to the payee and only with the payment service user's explicit consent;

(d) every time a payment is initiated, identify itself towards the account servicing payment service provider of the payer and communicate with the account servicing payment service provider, the payer and the payee in a secure way;

(e) not store sensitive payment data of the payment service user;

(f) not request from the payment service user any data other than those necessary to provide the payment initiation service;

(g) not use, access or store any data for purposes other than for the provision of the payment initiation service as explicitly requested by the payer;

(h) not modify the amount, the payee or any other feature of the transaction.

4. The account servicing payment service provider shall:

(a) communicate securely with payment initiation service providers;

(b) immediately after receipt of the payment order from a payment initiation service provider, provide or make available all information on the initiation of the payment transaction and all information accessible to the account servicing payment service provider regarding the execution of the payment transaction to the payment initiation service provider;

(c) treat payment orders transmitted through the services of a payment initiation service provider without any discrimination other than for objective reasons, in particular in terms of timing, priority or charges vis-à-vis payment orders transmitted directly by the payer.

5. The provision of payment initiation services shall not be dependent on the existence of a contractual relationship between the payment initiation service providers and the account servicing payment service providers for that purpose.

Article 80
Irrevocability of a payment order

1. Member States shall ensure that the payment service user shall not revoke a payment order once it has been received by the payer’s payment service provider, unless otherwise specified in this Article.

2. Where the payment transaction is initiated by a payment initiation service provider or by or through the payee, the payer shall not revoke the payment order after giving consent to the payment initiation service provider to initiate the payment transaction or after giving consent to execute the payment transaction to the payee.

3. However, in the case of a direct debit and without prejudice to refund rights the payer may revoke the payment order at the latest by the end of the business day preceding the day agreed for debiting the funds.

4. In the case referred to in Article 78(2) the payment service user may revoke a payment order at the latest by the end of the business day preceding the agreed day.

5. After the time limits laid down in paragraphs 1 to 4, the payment order may be revoked only if agreed between the payment service user and the relevant payment service providers. In the case referred to in paragraphs 2 and 3, the payee's agreement shall also be required. If agreed in the framework contract, the relevant payment service provider may charge for revocation.

In the case of unauthorised transactions the customer can address a refund claim to the banks, even where a PISP is involved, and the banks have immediately to indemnify him. Then the PISP is obliged to compensate immediately the bank where it is liable for an unauthorised payment transaction or a non-executed or defective payment. The burden of proof is anyhow on the PISP.

3.4.3 Regulatory Compliance

To comply with the new legislation, service providers, both banks and new TPPs, must follow some key steps. For the sake of clarity we divide the following description into three parts: in the first part we will describe what already existing Service Provider will have to do to fit the PSD2, in the second we will focus on PSD's exemptions and in the last one we are going to explain how should a new TPP be involved.

1. According to the art.109 of the New Directive the already authorized services provider will have to submit a report to the competent authorities certifying that they have all the requirements to continue to be PSPs. They must ensure that they have corporate governance arrangements and proportionate, valid and adequate internal control mechanisms; procedures for monitoring and managing security incidents, adequacy of capitalparticipants, honors and professionalism of company representatives,

minimum capital, professional insurance for AISP and PISP; segregation of their funds from those of customers where they are detained.

Article 109
Transitional provision

(a) Member States shall allow payment institutions that have taken up activities in accordance with the national law transposing Directive 2007/64/EC by 13 January 2018, to continue those activities in ac-cordance with the requirements provided without being required to seek authorisation or to comply with the other provisions laid down until 13 July 2018. Member States shall require such payment institutions to submit all relevant information to the competent authorities in order to allow the latter to assess, by 13 July 2018, whether those payment insti-tutions comply with the requirements laid down and, if not, which measures need to be taken in order to ensure compliance or whether a withdrawal of authorisation is appropriate. Payment institutions which upon verification by the competent au-thorities comply with the requirements laid down shall be granted authorisation and shall be entered in the European Central Registers. Where those payment institutions do not comply with the requirements laid down by 13 July 2018, they shall be prohibited from providing payment services.

(b) Member States may provide for payment institutions referred to in paragraph 1 of this Article to be automatically granted authorisation and entered in the European Central Registers if the competent au-thorities already have evidence that the requirements laid down are complied with. The competent authorities shall inform the payment institutions concerned before the authorisation is granted.

(c) This paragraph applies to natural or legal persons who have beneted from new information exchange arrangements before 13 January 2018 and exercised payment services activities. Member States shall allow those persons to continue those activities within the Member State concerned in accordance with PSD, until 13 January 2019 without being required to seek authorisation or, to obtain an exemption, or to comply with the other provisions laid down. Any person referred to in the first subparagraph who has not, by 13 January 2019, been authorised or exempted under this Directive shall be prohibited from providing payment services.

(d) Member States may allow natural and legal persons beneting from an exemption as referred to in paragraph 3 of this Article to be deemed to benet from an exemption and automatically entered in the European Central Registers where the competent authorities have evidence that the requirements laid down are complied with. The competent authorities shall inform the payment institutions concerned.

2. regards to service providers operating under exemption from the PSD, the new Directive estabilishes that:

> (a) Member States must authorize those who are already exempted from pursuing such activity in accordance with the PSD until 13 January 2019 without the need to submit a specific authorization to operate as a PSP;
>
> (b) any institution who has started its activity in accordance with the national transposing legislation of the PSD by 13 January 2018 and for which the authorization has not been granted, i.e. the possibility of operating under exemption, is prohibited from providing payment services in accordance with Article 37 of the New Directive.

Article 37

Prohibition of persons other than payment service providers from providing payment services and duty of notification

(a) Member States shall prohibit natural or legal persons that are neither payment service providers nor explicitly excluded from the scope of this Directive from providing payment services.

(b) Member States shall require that service providers for which the total value of payment transactions executed over the preceding 12 months exceeds the amount of EUR 1 million, send a notification to competent authorities containing a description of the services offered.
On the basis of that notification, the competent authority shall take a duly motivated decision where the activity does not qualify as a limited network, and inform the service provider accordingly.

(c) Member States shall require that service providers carrying out a payment in addition to a pre-existing subscription send a notification to competent authorities and provide competent authorities an annual audit opinion, testifying that the activity complies with the limits set out in this Directive.

(d) Notwithstanding paragraph 1, competent authorities shall inform EBA of the services notified pursuant to paragraphs 2 and 3, stating under which exclusion the activity is carried out.

(e) The description of the activity notified under paragraphs 2 and 3 of this Article shall be made publicly available in the European Central Registers (EBA Registers).

(f) Member States may allow persons acting on the basis of one of the exemptions provided for in the PSD to be exempted and automatic-ally entered in the EBA Registers where the competent authorities have evidence that they comply with the requirements.

3. those who want to get PSP for the first time will have to:

   (a) submit an authorization, together with the information already provided for in the current supervisory arrangements, in which they will have to be more detailed references to:

      (I) existing procedures for monitoring and managing safety accidents and complaints;

      (II) provisions on operational continuity;

      (III) the principles and definitions applied to the collection of statistical data in relation to transactions and fraud;

      (IV) a statement on security policy.

   (b) undertake professional liability insurance, or provide other similar guarantee, if the applicants intend to submit a request for the provision of payment order services or information on accounts. Follow-ing authorization by the competent national authority, details of the newly authorized payment institutions will be reported first in the individual national currency and subsequently in the EBA Registers.

Furthermore, the National Government will have to provide specific indications referring to the rights and obligations placed on each part in the relationships between the new payment service providers, the individual payers and the payment service providers where the accounts have been opened while at the same time regulating the liability regime for each.

3.5 Strong Customer Authentication

SCA is an authentication system used in the modern applications and devices. In this paragraph we are going to introduce briefly this system and then we will define some technical standards.
SCA is introduced to intensify the customer's control above his/her own applications.
The Strong Customer Authentication must provide an authentication system based on at least two of the three following type of security factors:

- **Knowledge**: This is the most common type of access system. It is based on the use of something that only the customers knows as for example a username and a password or a PIN code;

- **Possession**: Also this type is widely diffused. It consists of the use of a code that the service provider sends to the customer at the moment of the authentication;

- **Inherence**: The last one is the most recent type of access. It is based on the recognition of the person who is authenticating. An example of this feature is the fingerprint or the facial recognition.

3.5.1 Regulatory Technical Standards

On August 2016 EBA has issued the Regulatory Technical Standards (RTS) under Strong Customer Authentication (SCA) and Open and Secure Communications Standards. All this goes on to address the technical issues underlying the PSD2 directive, incorporating the comments and feedback so far received by stakeholders but also suggesting new questions in consultation on issues still open.
There are also clarifications as:

1. requirements for identifying and securing the banking, PISP and AISP communication sessions based on authentication certificates and encrypted messages;

2. payment security requirements;

3. exemptions from adoption to SCA;

4. the need to align the information provided to AISP by the bank;

5. treatment of sensitive customer data.

3.5.2 SCA and exceptions

The requirement to apply the SCA involves the whole system of payments, in particular SCA must be implemented by PISP, AISP and banks.
SCA is an authentication system based on two or more factors as knowledge (i.e. something only the customer knows, as for example a password or a pin), possession (i.e. something only the customer possesses, as for example a token), and inherence (i.e. something the customer is, as for example the fingerprint or the iris scanning).

The PSD2 requires SCA when the user:

- accesses to his/her payment account online;

- initiates an electronic payment transaction;

- does anything else that may present a risk of payment fraud or other abuses.

In some cases, the PSD2 provides the possibility of not using the SCA:

- Trusted Beneficiary, i.e. the outgoing payments to someone included in a white lists for that customer;

- Transactions from the customer to him/herself , i.e. the transactions between two accounts of the same customer held at the same bank;

- Low risk transactions, i.e. payments within the same services provider justified by a transaction risk analysis;

- Low value payments, i.e. the transactions not exceeding 30 € or transactions made with a payment instrument that has a spending limit of 150 €;

- No sensitive payment data, i.e. all the purely consultative services without showing sensitive customer or payment information.

The use of SCA is governed by Article 97 of PSD2:

Article 97
Authentication

1. Member States shall ensure that a payment service provider applies strong customer authentication where the payer:

(a) accesses its payment account online;

(b) initiates an electronic payment transaction;

(c) carries out any action through a remote channel which may imply a risk of payment fraud or other abuses.

2. With regard to the initiation of electronic payment transactions as re-ferred to in point (b) of paragraph 1, Member States shall ensure that, for electronic remote payment transactions, payment service providers apply strong customer authentication that includes elements which dynamically link the transaction to a specific amount and a specific payee.

3. With regard to paragraph 1, Member States shall ensure that payment service providers have in place adequate security measures to protect the confidentiality and integrity of payment service users' personalised security credentials.

4. Paragraphs 2 and 3 shall also apply where payments are initiated through a payment initiation service provider. Paragraphs 1 and 3 shall also apply when the information is requested through an account information service provider.

5. Member States shall ensure that the account servicing payment service provider allows the payment initiation service provider and the account in-formation service provider to rely on the authentication procedures provided by the account servicing payment service provider to the payment service user in accordance with paragraphs 1 and 3 and, where the payment initiation service provider is involved, in accordance with paragraphs 1, 2 and 3.

3.6 Data Protection

The PSD2 also establishes a set of regulations that various actors (AISP, PISP and banks) have to follow in handling sensitive data.
Quoting the law, art.4(32): "Sensitive payment data means data, including per-sonalised security credentials which can be used to carry out fraud. For the activities of payment initiation service providers and account information ser-vice providers, the name of the account owner and the account number do not constitute sensitive payment data."
In the continuation of the discussion, for the sake of clarity, we will divide this section into two parts: in the first part we will analyze how the PISP should behave, in the second we will do the same with the AISP.
In each of the 3 paragraphs that follow, we will describe the duties of TPPs, banks and users.

3.6.1 PISP

First of all the PISP has to guarantee that any other information about the customer obtained during the payment initiation, is provided only to the payee after the customer's explicit consent. Furthermore it can't store customer's payment sensitive data under any form or request information to the customer that is not necessary for initiation of the payment. Then the PISP must not use, access or store data for other purposes different from providing payment initiation as explicitly requested by the customer. Finally the PISP has the duty not to modify the amount, the payee or any other information of the transaction. On the other hand the bank shall provide and make available to the PISP all information on the initiation of the payment, right after receiving the payment order from it.

3.6.2 AISP

The AISP has to provide services only following the customer's explicit consent because the service may vary according to the will of the customer. Moreover, it has to access only the information from designated payment accounts and associated payment transactions and hasn't to request sensitive payment data linked to the payment accounts. Finally, it is very important to remind that it's mandatory for the AISP not to use, access or store any data for purposes different from performing the services explicitly requested by the customer, in accordance with data protection rules.

Article 94
Data protection

1. Member States shall permit processing of personal data by payment systems and payment service providers when necessary to safeguard the prevention, investigation and detection of payment fraud. The provision of information to individuals about the processing of personal data and the processing of such personal data and any other processing of personal data for the purposes of this Directive shall be carried out.

2. Payment service providers shall only access, process and retain personal data necessary for the provision of their payment services, with the explicit consent of the payment service user.

3.7 EBA Regulatory Technical Standards

The European Bank Authority released the Regulatory Technical Standards for authentication and communication between TPPs, banks and users.The EBA RTS was published on February 2017 and it has to be applied in 18 months.
It specifies:

- requirements of SCA;
- requirements to protect user's personal credentials;
- requirements for communication and identification between users, TPPs and banks;
- exemptions to the application of SCA, based on the level of risk for the services provided, the amount of the transaction, the payment channel.

This regulatory should ensure a high level of security to the banks, the users and the TPPs, the safety of users' funds and it should also allow the development of innovative and user-friendly means of payment.
Furthermore it indicates the requirements of common and open standards of communication, so the EBA RTS should:

- allow the online payment;
- guarantee the technological multi-platform interoperability;
- ensure that the bank is aware that it is being contacted by the TPPs and not by the customers;
- ensure that the TPPs is communicating with the banks in a safe way.

To be compliant with the EBA RTS on authentication and communication, the PISP and the AISP must identify themselves to the bank every time they provide their services, payment initiation, aggregating information or confirmation on the availability of funds, and every time they communicate to the bank, to the payer, to the payee in a secure way in accordance with article 98. Obviously also the bank must ensure a safe communication with the TPPs.

Article 98
Regulatory technical standards on authentication and communication

1. EBA shall, in close cooperation with the ECB and after consulting all relevant stakeholders, including those in the payment services market, reflecting all interests involved, develop draft regulatory technical standards addressed to payment service providers as set out in Article 1(1) of this Directive in accor-

dance with Article 10 of Regulation (EU) No 1093/2010 specifying:

(a) the requirements of the strong customer authentication referred to in Article 97(1) and (2);

(b) the exemptions from the application of Article 97(1), (2) and (3), based on the criteria established in paragraph 3 of this Article;

(c) the requirements with which security measures have to comply, in accordance with Article 97(3) in order to protect the confidentiality and the integrity of the payment service users' personalised security credentials;

(d) the requirements for common and secure open standards of communication for the purpose of identification, authentication, notification, and information, as well as for the implementation of security measures, between account servicing payment service providers, payment initiation service providers, account information service providers, payers, payees and other payment service providers.

2. The draft regulatory technical standards referred in paragraph 1 shall be developed by EBA in order to:

(a) ensure an appropriate level of security for payment service users and payment service providers, through the adoption of effective and risk-based requirements;

(b) ensure the safety of payment service users' funds and personal data;

(c) secure and maintain fair competition among all the providers of payment service;

(d) ensure technology and business-model neutrality;

(e) allow the development of user-friendly, accessible and innovative means of payment.

3. The exemptions referred in point (b) of paragraph 1 shall be based on the following criteria:

(a) the level of risk involved in the service provided;

(b) the amount, the recurrence of the transaction, or both;

(c) the payment channel used for the execution of the transaction.

4. EBA shall submit the draft regulatory technical standards referred in paragraph 1 to the Commission by 13 January 2017.
Power is delegated to the Commission to adopt those regulatory technical standards in accordance with Articles 10 to 14 of Regulation (EU) No 1093/2010.

5. In accordance with Article 10 of Regulation (EU) No 1093/2010, EBA shall review and, if appropriate, update the regulatory technical standards on a regular basis in order, inter alia, to take account of innovation and technological developments.

3.8 A global overview

To conclude this section we summarize the changes made by PSD2:

Bank: The Bank is the central institute in the economy, full-trusted by customers. Before PSD2 the bank is the only protagonist of the financial system. With PSD2 Banks open totally to TPPs, both about information management and payments initialization, making the customer a more central figure of the whole system. In any case, the bank keeps control through the design of APIs to provide the TPPs.

SCA: SCA is an authentication protocol based on two factors selected from the three possible: Possession, Knowledge or Inherence. It is mandatory to implement it by the TPPs, especially for those who make dispositive transactions.

AISP: It is an aggregator of bank account information, it must be able to get information from multiple banks at the same time and to put them together in a more convenient and user-friendly way. It's important to remind that the AISP can only carry out informative and non-dispositive operations and it must ensure the use of SCA with the customer, in addition to being registered within the EBA Registers..

PISP: It is a payment service provider that acts as an intermediary between the customer and its account providing the impetus to the payment. As for AISP, PISP must ensure the use of SCA with the customer and must be registered within the EBA Registers. Any institution, even if it is not a bank or similar, can register as PISP to the central authorities and be registered in the EBA Registers. The PISP should only deal with dispositive operations (payments, etc..) without directly access to the customers' account.

4. PSD2 & BLOCKCHAIN: AN IMPLEMENTATION MODEL

On the basis of what we have seen so far we sought a way to implement the technology of Blockchain in the context of the PDS2. The major problems were tied to the different context in which the Blockchain was born: in the mechanism of Bitcoin the Blockchain plays a role of both shared database as well as an instrument for the validation of the transactions, with the system of proof-of-work exhibited previously. In a scenario in which the nodes of the network would be represented by European banks, it was unthinkable structuring the Blockchain in a manner similar to that used for Bitcoin, for a series of substantial differences: trusted nodes, the need to maintain the management of risk in the hands of the banks, much quicker. Therefore the model of Blockchain had to be reshaped in such a way as to maintain all the advantages of a distributed database but without causing defects of an instrument for public validation.

On the basis of these observations we propose our deployment model based on Blockchain wherein the fundamental aspects of the issue are the hybrid setting of the model, the absence of proof-of-work and the presence of a privileged, but not authoritarian node.

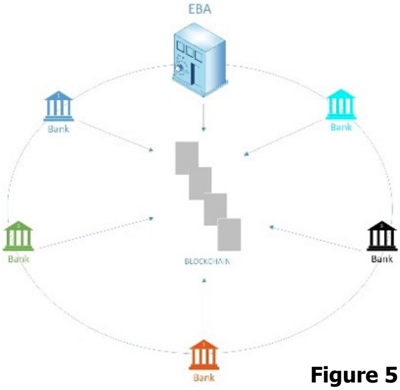


Figure 5: Blockchain proposal

4.1 Blockchain Proposal

The presence of trusted and certificated nodes and the need to maintain a certain level of privacy without compromising the sharing, prompted us toward the choice of a hybrid Blockchain, wherein access and reading permissions are granted only to trusted entities and already known to the network, while only and exclusively the trusted nodes can write on Blockchain. The choice of a hybrid Blockchain requires the presence of a privileged node, a node able to play the role of certification authority without however having sufficient authority to change the outcome of a validation. In our model the EBA plays this role.

In Bitcoin the validation of a new block potentially involves every node of the network. This aspect is crucial for ensuring the sharing and the transparency of the whole structure, and therefore also in our model, the validation process requires the intervention of each node. Remember that from now on every time that we speak of nodes we intend a bank. Figure 5 shows how Banks and EBA should collaborate in the creation and management of the Blockchain.

The determinant aspect concerns the validation process: transactions will not be authorized by the Blockchain, as this would entail a waste of time compared to today's processing time as well as a loss of authority on the part of banks, but will be checked by means of a process of verification of the digital signature. We will see in detail how and when this verification occurs at the moment we remark the substantial difference between our Blockchain and those currently used for the cryptocurrencies: in our model the risk management and the interbank network remain unchanged, that is renewed is the database on which to store data and access methods.

Finally each block will contain transactions from multiple banks. We will see later the process of creation of a new block, but we want to remark from now the importance of the cryptographic aspect to ensure both accessibility and privacy to a database of these dimensions.

4.1.1 Structure of a Block

The main difficulty that we found was to build a Blockchain functional to mechanism and timing of banking, that maintains the basic structure of a distributed database but that allows further operations such as for example read access by third parties, if authorized. These difficulties are reflected directly on the structure of the Blockchain, more precisely on the contents of each single block, as a model similar to Bitcoin fails to guarantee all those features required both by the PSD2 is intrinsically by the banking.

Based on what has been said so far, and on Bitcoin example, we propose a Blockchain composed of blocks with this structure (figure 6):

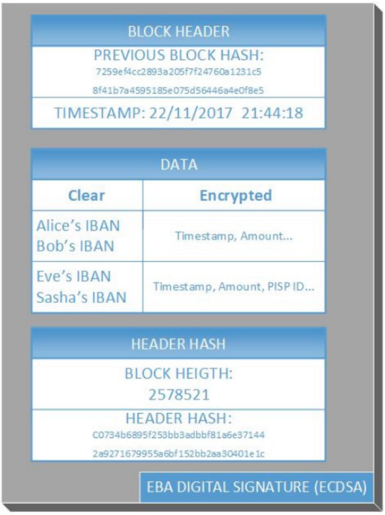

BLOCK HEADER
PREVIOUS BLOCK HASH:
7259ef4cc2893a205f7f24760a1231c5
8f41b7a4595185e075d56446a4e0f8e5
TIMESTAMP: 22/11/2017 21:44:18
DATA
Clear
Encrypted
Alice's IBAN
Bob's IBAN
Timestamp, Amount...
Eve's IBAN
Sasha's IBAN
Timestamp, Amount, PISP ID...
HEADER HASH
BLOCK HEIGTH:
2578521
HEADER HASH:
C0734b6895f253bb3adbbf81a6e37144
2a9271679955a6bf152bb2aa30401e1c
EBA DIGITAL SIGNATURE (ECDSA)

Figure 6: Structure of a block

We provide an explanation of the three sections that compose the block:

- **Block Header**

It contains the hash of the previous block to which it is connected and all the technical information of the block, like the timestamp of its creation.

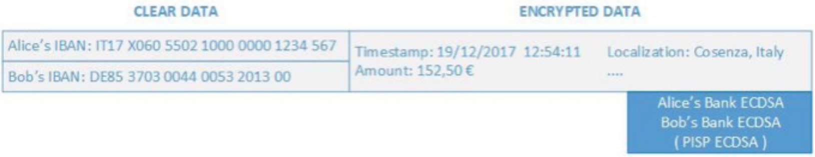
- **Data**

It is the heart of the block; this section is intended to contain the transactions. Each datum is divided in two parts, a clear one containing the IBAN of customers (or its token) and an encrypted one containing all the sensitive data of the transaction, as the sum of money involved or the geolocalization. The number of transactions stored in a block can be fixed but it depends on multiple factors that we can't estabilish, so we will call this number N.

- **Header Hash**

The third section is perhaps the most important; it contains the block height (the number of block in the chain) and especially the hash of the block which is calculated after the last transaction was inserted. The Hash guarantees the immutability of the block, because the slightest modification of data would cause a radical change of the string of hash, causing a decoupling between the block in question and the subsequent.

We now analyze in detail the Data section, specifically the structure of every single transaction.



CLEAR DATA	ENCRYPTED DATA	
Alice's IBAN: IT17 X060 5502 1000 0000 1234 567	Timestamp: 19/12/2017 12:54:11 Amount: 152,50 €	Localization: Cosenza, Italy ....
Bob's IBAN: DE85 3703 0044 0053 2013 00		
		Alice's Bank ECDSA Bob's Bank ECDSA (PISP ECDSA)

We have already outlined the dual nature how transactions are saved in the block, the first non-encrypted part consisting ofthe identifiers of the customers involved only and the encrypted part containing the rest of the data. This choice may seem unadvised but it was necessary as the PSD2 requires ensuring the data access to TPPs. This subdivision in fact solves a thorny problem: how to allow an AISP to obtain information regarding its client avoiding long times and the problems related to the management of the keys.

Suppose in fact the following scenario: a fully-encrypted Blockchain (maybe with a public-key system) and an AISP that must carry out a research about a client. The only solution would be to scan (and decrypt) the Blockchain completely and then collect the data related to the individual customer. The time required to complete a total scan would be incredibly long; also the addition of the time needed for deciphering the operation would make it further slowly and expensive already for a single customer, which is unacceptable for the European scenery.

To avoid a similar situation we have divided the data section as follows: AISP should only scan non encrypted sections to research the identifier corresponding to the customer and subsequently decrypt only the data that will be collected, guaranteeing privacy, since the data relating to other customers will not be decrypted, as well as efficiency.

The presence of the two (or three) Elliptic Curves Digital Signature Algorithm is functional to the validation process (it will be discussed in the next section). These are calculated by the Banks involved and by the eventual PISP once the transaction has been authorised by the Interbank network. Of course, the cryptographic question remains an open problem; in the months in which we have worked on this project we have made several assumptions of cryptosystem without reaching a final solution; we are convinced that by drawing from protocols already existing, such as SSL and inserting more advanced mathematical techniques, as elliptic curves, we could build the necessary cryptographic system. It is our intention to continue this research in collaboration with several Italian Universities.

4.1.2 Validation Process

In this section we will look into the matter of consensus and validation of a block, in fact there can be no talk of Blockchain, or in general of distributed ledger, without touching the topic of consent.

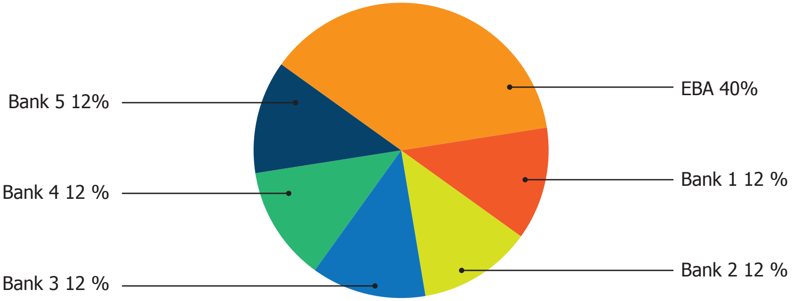
Let us briefly analyze the validation process in Bitcoin: after a transaction is broadcasted to the Bitcoin network, it may be included in a block that is published to the network. When this happens it is said that the transaction has been mined at a depth of 1 block. With each subsequent block that is found, the number of blocks deep is increased by one. To be secure against double spending, a transaction should not be considered as confirmed until it is a certain number of blocks deep. The classic Bitcoin client will show a transaction as "unconfirmed" until the transaction is 6 blocks deep. Merchants and exchanges who accept bitcoins as payment can and should set their own threshold as to how many blocks are required until funds are considered confirmed.

Remember that for "Mining" we mean the resolution of a very complex cryptographic puzzle, which requires a huge number of calculations to be resolved, and therefore of time.

In the world of interbank transactions time is precious, the rapidity of exe-

cution of a payment is a fundamental element for the user experience and cannot be bartered, even for the more secure system of the world. A secure system that takes 6 minutes to authorize payment would cause discomfort even in small things of every day, such as shopping or the payment of a bill.
Based on these assumptions we have conceived a model of Blockchain that eliminates the long waiting times without compromising the advantages of a distributed consensus. The consent in fact will no longer be issued on the goodness of a transaction, this in fact is and remains the exclusive responsibility of the bank but must be a formal approval of the work of the Bank.
Given this premise it is clear at this point what role digital signatures play in our model: they serve as a guarantee for every single transaction, a seal positioned by the nodes involved to ensure the customer and the entire network that the actions undertaken in that transaction were properly authorized. It is clear that a similar mechanism cannot be applied in the Bitcoin model, since the nodes are not trusted and above all there are no certificates or guarantees; indeed a network consisting of banks has the enormous advantage of having full-trusted nodes that allows us both to eliminate the costly mechanical of Proof-of-work and to exploit the solid nature and reassuring the bank.
Thus we see an example of the validation in our model to understand concretely who and how ensures the consent. Suppose a scenario consisting of the EBA (privileged node) and 5 banks where the EBA has a decisional powers of 40% and the banks will share equally the remaining 60%; a block is validated only after having exceeded a threshold of the validation of the 50%.

Figure 8: Pie chart of validation



Of course the real-world scenario would be very different, however in our example we wanted to stress a fundamental aspect: the first concerns the weight of decision-making of the EBA, it must in fact have a higher percentage with respect to a bank since it must play the role of certifying autorithy, but not high enough to affect autonomously the outcome of a decision (thus avoiding the defects of the Blockchain public and private).

The validation process includes the following steps:

1. Once a transactions is validated, it is signed by Banks (and eventually by PISP) involved and it is sent to EBA's Server. EBA checks only Digital Signatures and inserts the transaction into the new block. If the control detects inconsistencies the transaction is momentarily frozen and returned for a subsequent control.

2. After N transactions have been inserted, EBA creates the new block, calculates its Hash and signs it digitally. The block is sent to the network to be validated.

3. Each Bank checks the validity of the block, verifying only the digital sig-natures and Hash.

4. When the block reaches the validation threshold, it is inserted into the Blockchain by EBA.

The numbers that we have used here are purely formal, the threshold of approval to 50%, the number N of transactions, the decisional powers of the EBA are all key factors of the model that should and can be decided only by having a clear and comprehensive overview of the context of European Banking.

Table 2: Final considerations

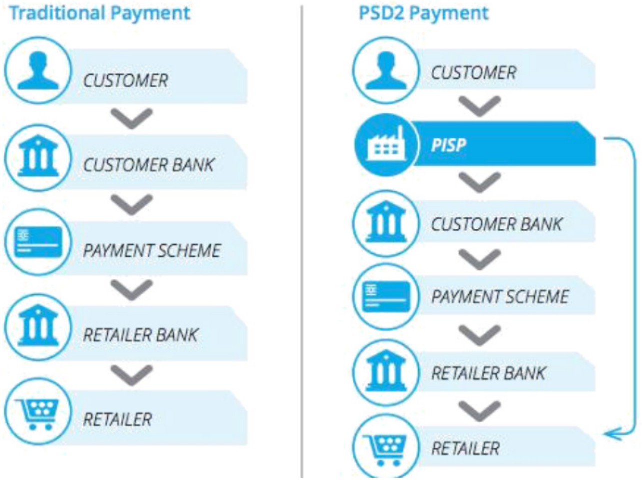
Pros	**Cons**
Data Integrity	Need an improved hardware solution
Advanced cryptographic tools	Need an internal reorganization
Replicated copies	No central authority
Resilient to malicious data manipulation	
Transparency & privacy	
Legal Validity	

4.2 PISP Model

This section is devoted to the description of the PISP (Payment Initiation Service Provider); for the sake of clarity we remind that it is defined as a communication channel between a customer and his/her bank. In order to understand the proposed model we propose an ideal scenario composed by a customer, a retailer and two banks: the customer's bank and the retailer's bank. Suppose now, due to a purchase, the existence of a transaction betwe-

en the customer and the retailer. We will describe the following transaction from four different points of view, each of which can be considered as a different layer associated to different degrees of abstraction.
The process layer is necessary to describe what kind of information are ex-changed among the entity (e.g. customer,retailer, and banks) when a transaction starts.

In the logical layer the set of data necessary to ensure the feasibility of the transaction is described.
Next, the infrastructure layer contains the most concrete details about the adopted infrastructure, cryptography scheme, messages fields, and adopted protocols.
Finally, the last layer is dedicated to the description of a possible case use.



4.2.1 Process layer

In Figure 9 the main steps of the proposed model are described and compared with respect to the current payment scheme considering the same scenario composed by a customer, a retailer, and their banks. A first observation is about the entities involved in the payment process, the unique change is about the information flow and the introduction of the PISP entity who has the fundamental task to connect the customer with his/her bank and to initiate the payment pro-cess. The PISP will should not receive or manage the funds of the customer, it is limited to checking whether the payer has sufficient funds on his/her bank account in order to complete the transaction by means of preventive secure authentication. The rest of the transaction life-cycle is unchanged, the unique add-on is represented by the introduction of the PISP and its interaction with the payer and the bank.

4.2.2 Logical Layer

In referral to the just mentioned scenario composed by the customer, the retailer, and their banks, we want to analyse what kind of data is exchanged among the entities during the transaction life-cycle from a more detailed point of view. Moreover, also in this case, we compare the proposed model with respect to the payment scheme currently in use.
As also highlighted in the description of the process layer, the direct relation between the customer and the PISP is evident.
The first step in the transaction life cycle is actuated by the customer,by sending a payment instruction and/or authorization to the PISP. In this step an identifier about the customer, the retailer, and the other transaction details must be transmitted to the PISP. On the basis of the received information the

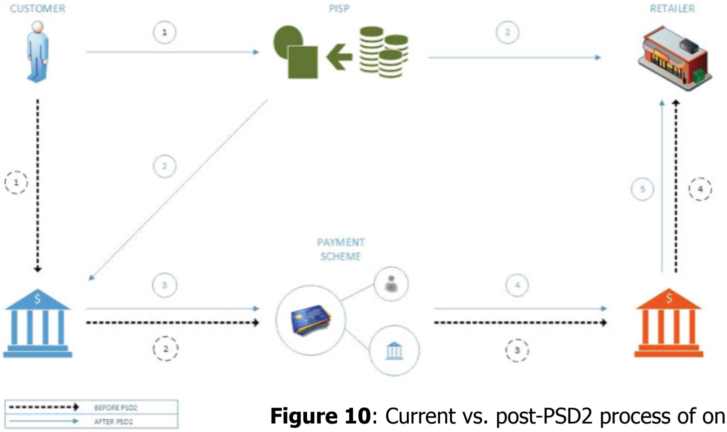


Figure 10: Current vs. post-PSD2 process of online payment with debit/credit card

PISP is able to identify which banks are involved in the payment transaction and it opens a connection with the customer's bank. The PISP is not interested to other details of the transaction, only the identification of the banks involved in the process is mandatory. Once the customer's bank is identified, the PISP sends a payment instruction thanks to the APIs provided by the ASPSP (customer's bank). When the ASPSP receives the transaction, it verifies the integrity and the validity of the received message and sends a feedback notification to the PISP. In case of an authorised transaction the PISP can notify the retailer's bank about the validity of the transaction.

Since this point the transaction life cycle is untouched respect to its actual implementation in terms of adopted protocols and standards. In this phase PSD2 suggests to adopt the interoperability system standard ISO 20022 about the definition of more than 300 standard messages.

The last step is completely demanded to the bank and it corresponds to the archiving of the transaction. The introduction of the new payment directive represents an opportunity to improve the archiving systems actually adopted by the ASPSPs. To this end, in the proposed model, we suggest to adopt new solutions based on the distributed ledger technology. More details about the adoption of this kind of solution will be described in the architecture layer.

The PSD2 demands to the PISP also the task to implement a procedure about a refund of payment. In this case use the customer reports an unauthorised payment to his/her ASPSP, then the bank restores the debit payment account and sends a refund request to the PISP. The PISP compensates the losses of ASPSP and, once the refund is complete, the bank archives the transaction. With the aim to provide a clear description of the logical layer, we analyse the new interfaces between the main entities of the proposed model by defining the type of messages exchanged during the transaction life cycle.

- **Customer - PISP**: The PISP task consists in the initialization of the payment procedure, in this phase it is necessary to communicate to the PISP all the details about the transaction (e.g. customer and retailer accounts numbers, amount, etc). The unique data directly used by the PISP is the customer account, due to this reason we suggest in our model the adoption of a message unreadable from the PISP perspective. In this way the PISP is unable to comprehend the data in the message but is able to forward the payment to the ASPSP via APIs.

- **PISP - ASPSP**: In this phase the message forwarded by the PISP to the ASPSP must be completely readable from the bank in order to complete the transaction.

The details about the encryption schemes adopted in these communications will be detailed in the description of the architecture layer.

4.2.3 Architecture Layer

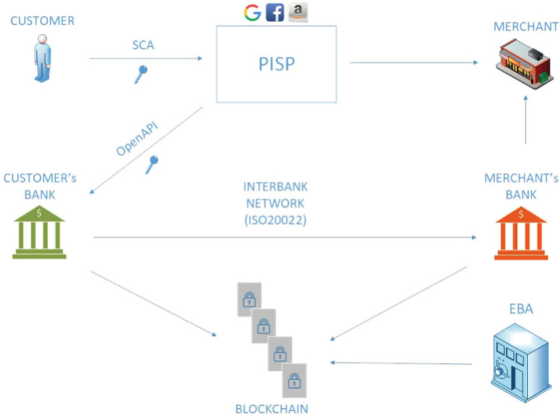
Now we can take a look to the details about the proposed architecture for the PISP implementation service. As dened by the logical layer, the architecture has to guarantee the management of two interfaces: Customer - PISP and PISP - ASPSP. Looking at the proposed system from a physical perspective we have designed these two interfaces as two data exchange sessions between the customer (PSU) and a couple of web services. The first session is devoted to the customer authentication by respecting the requirements of strong customer authentication: in this case the PISP is working as a communication channel between the PSU and the ASPSP authentication servers.

The second session is the real transsaction:, in this case the PISP works by ensuring a connection between the PSU and the ASPSP resource server. In this implementation the adoption of web services and the open APIs is fundamental to guarantee to the AISP the access to the ASPSP private data. The data in the PISP's model has to go from the customer to the bank, through the PISP, in a secure way. To make it possible we suggest to use a private key to encrypt data. This private key must be shared between PSU and ASPSP.

4.2.4 PISP Use Case

• **Step 1 - Request Payment Intitation**: This flow begins with a PSU consenting to a payment to be made. The request is sent through a PISP. Before sending the request, the customer has to authenticate to the PISP through the SCA if it determines that none of the SCA exemptions apply. The debtor account details can optionally be specified at this stage.

• **Step 2 - Setup Single Payment Initiation**: The PISP connects to the ASPSP that services the PSU's payment account and creates a new payments resource. This informs the ASPSP that one of its PSUs intends to make a payment and it has to authorize the payment if all the conditions are respected. This step has to be regulated thanks to the OpenAPI.



- **Step 3 - Interbank Payment Network**: The customer's bank contacts the merchant's one through the already existing interbank network. So the payment can be done as actually is done, respecting the rule of ISO 20022. In this way the risk management is always handled by the banks. After the payment has been carried out, the merchant's bank informs its customer in such a way to authorize the purchase.

- **Step 4 - Data Entry in the Blockchain**: Until this step the parts involved have no contact with the Blockchain. Just after the authorization of the transaction, the two banks encrypts the data of the transaction as described before and then they send it to the European Banking Association in such a way to enter it in the Blockchain.

4.3 AISP Model

This section is devoted to the description of the AISP. We remind that the AISP is a service provider, chosen by the user, which aggregates information from every bank's blockchain, where the customer has an account. To clarify the whole process, proposed in our model, we will describe an ideal scenario composed by a User, an AISP and one or more Banks' Blockchains. Moreover, we suppose that the Banks make available some APIs that should permit to the AISP to read the Blockchain and collect data from it. Also in this section, we will divide the model into four different points of view: the process layer, the logic layer, the architecture layer and finally an example of case use.
In the following figure the AISP's model that we propose in this paper is represented.

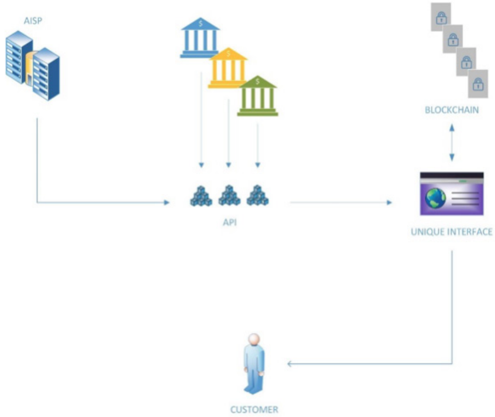
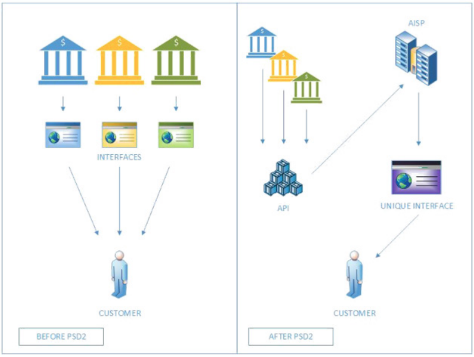


Figure 12: The proposed model

4.3.1 Process Layer

In the continuation of the paper, we will describe and compare this model and all its steps, comparing to the already existing scenario. (fig.13)
The existing scenario is really ramified: first of all the customer has to relate with various interfaces, as for example the home-banking, different for every bank. It currently doesn't exist a service that allows the customer to handle more than one bank account together, so the customer hasn't a view of his/her whole balance sheet or has to aggregate data and informations obtained from multiple interfaces by him/herself. Instead a TPP will be created after PSD2 for this purpose, the AISP. After the introduction of the AISPs, the customer has a powerful tool that allows him/her to handle all the bank accounts, relating directly with all the bank databases. The AISPs will use the APIs provided by the banks to create an unique interface that allows the user to access to the blockchain of each bank, these APIs will have the same tasks as existing ones, but probably they will have to be designed in a different way.



It's important to remind that the AISP can't take part to any order transactions, but it can just study the habits and the behaviour of customers to meet their needs and to anticipate their preferences.
Finally, the user will probably choose the AISP, or more AISPs if he/she wants, based on the interfaces, the services offered, how the information is aggregated and how much user friendly the AISP is.

4.3.2 Logic Layer

In this subsection we are going to describe in detail the set of data necessary to allow the exchange between the user, the AISP and the bank to ensure the feasibility of the model in fig.12.

First of all, there is a preliminary step to describe: the customer has to choose one, or more, AISP based on the services offered and his /her preferences. When they are making the contract, the customer must provide the IDs, provided by the banks, thus allowing the AISP to request information to each bank, where the customer has an account. For each banking account it's important that the user is able to customize the access level demanded to the AISP.

After that, the AISP initiates a request of information to the bank, providing the IDs of the customer. From now on we suppose that the AISP sends the request to a single bank. When the bank receives the request, it must verify if the AISP has been authorized; for this purpose it has just to check the IDs. Moreover, it's important that any unauthorized AISP cannot have access to the customer's data even if it has the IDs, for example provided at the time of signing a contract that is no longer valid. With this aim we propose to the bank to store the AISP authorized for each customer in a database and remove an AISP when a contract expires for any reason. Finally, the bank has to archive all the requests of information in the blockchain to ensure transparency and have a history of access to the accounts made by the AISPs.

When the bank verifies the request, it has to provide to the AISP all the information on the customer's account stored in the blockchain without aggregating them in anyway, because it is the main task of the AISPs. So, at this time, the AISP has to process all the information received by the banks and aggregate them in a user friendly way. Only after this step, the AISP provides information to the customer through its interface.

This process simplifies the search for customer's account information allowing the relation with an unique interface, provided by the AISP.

4.3.3 Architecture Layer

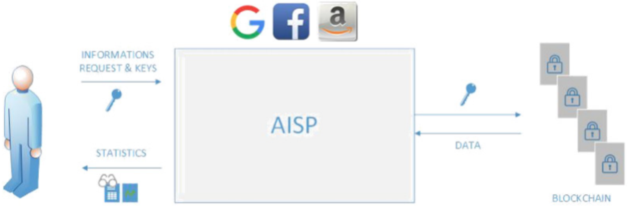
The last layer is devoted to a detailed description of the model architecture; therefore we are going to describe which solutions we propose to make feasible the whole process previously described.

Before starting the description it's important to remind that an AISP must obtain an eIDAS certificate to be recognizable as a valid AISP. This is necessary because the EBA wants to create a PKI to allow AISP, PISP and banks to communicate safely. After obtaining this certificate the AISP can enter its own product online at the appropriate stores.

The AISP has to be able to decrypt the data stored in the Blockchain, so it must obtain the decryption key for any PSU's transaction. Obviously it's mandatory for the AISP to not save keys to avoid the possibility of an unauthorized access. To make it possible it's suggested the use of a homomorphic encryption, or on the other hand the design of a timely API. In the continuation of the paper we are going to describe the various steps of the model described above (g.12) supposing to describe a scenario where the customer chooses and customizes a unique AISP.

4.3.4 AISP Use Case

• **Step 1 - Request Account Information**: This ow begins with a PSU allowing an AISP to access account information data. The authorization has to be regulated by the SCA. At the moment of the request, the PSU has to provide the tools to decrypt the data stored in the Blockchain.



• **Step 2 - Setup Account Request**: The AISP connects to the Blockchain where all the PSU's transactions are stored. Through the OpenAPI it finds and decrypts only the exact transactions. In this step some particular limitations to the AISP collection of information can be specified, as for example:

- Permissions - a list of data clusters that have been consented for access;

- Expiration Date - an optional expiration date for when the AISP will no longer have access to the PSU's data;

- Transaction Validity Period - the From/To date range which species a transaction history period which can be accessed by the AISP.

• **Step 3 - Data Collection and Elaboration**:

After the AISP obtains all the PSU's data requested and stored in the Blockchain, coming from all its multiple bank accounts, it has to aggregate them. This is the most important step for the AISP because it has to provide the information to the customer in the most user friendly way. So the AISP can create graphics, show all the balance sheets or aggregate them, advise investment or purchases in general based on the transactions made in the last period.

5. CONCLUSIONS

The18th of January 2018 is a date that must be marked in all the calendars. Indeed, the introduction of the new European payment directive could change the financial world as we know it.
PSD2 is a great opportunity for the parts involved, but they have to solve some open problems as for example how to satisfy the required security level or to share data without sharing sensitive customer's data.
As we suggested in this paper, our model could be a good solution to the problems raised by the European Directive.
The model uses a particular Blockchain, as described in the paper, that must be shared between EBA and the banks to solve the data storage problem; furthermore the cryptosystem it's desirable to be homomorphic such in a way to permit to work on it without decrypting to the other parts involved, as for example the AISP. Obviously this new technology will solve any data sharing problems. Finally the collaboration between the banks, the TPPs and the customers has to be handled by the OpenAPI, in order to use an existing technology in a different way.
The proposed Model can guarantee, thanks to the use of the described Blockchain, data integrity through its structure which uses hash functions and digital signatures. Moreover the data sharing permits to the parts involved to trust each other and to hold their own copy of Blockchain, in such a way that every malevolent action of a node or a hacker can be easily identified and opposed. The model also guarantees transparency and privacy, because the data will be shared between all the parts involved even if they are encrypted, so if for any reason a bank needs to read a certain transaction it should only ask for the key to that single transaction to the affected banks or the EBA. Finally, a legal validity of the model could be recognized because, as we just said, data stored in the Blockchain must be digitally signed by the EBA or the banks involved in the transactions, so this data can't be disavowed.

On the other hand, as for any new model, it is mandatory an in-depth research before its implementation. The research must be divided in two main topics: the need of an improved hardware solution, in order to make the model scalable, and the need of an internal reorganization, because it will be necessary as it will be mandatory for a new branch of each company involved to take care of this model. But there are also other cons, indeed the technologies exploited, as for example the homomorphic encryption, are innovative. This may be a problem, because it's necessary to study this before the application of the model, but this can be also a pro, in fact they can guarantee an higher level of security. We hope that this research can be as quick and fruitful as possible.

References

1. Andreas M. Antonopoulos Mastering Bitcoin, O'Reilly Media, Inc., First Edition, 2014.

2. Douglas Stinson, Cryptography: Theory and Practice, , Chapman and Hall, Third Edition, 2006.

3. Zachariadis, Markos, and Pinar Ozcan. "The API Economy and Digital Transformation in Financial Services: The Case of Open Banking" (2016).

4. Guibaud, Sophie. "How to develop a protable, customer-focused digital banking strategy: Open banking services and developer-friendly APIs." Journal of Digital Banking 1.1 (2016): 6-12.

5. Mansfield-Devine, Steve. "Open banking: opportunity and danger." Computer Fraud & Security 2016.10 (2016): 8-13.

6. European Commission Payment services (PSD2) Directive (EU) , https://ec.europa.eu/info/law/payment-services-psd-2-directive-eu-2015-2366.

6. APPENDIX A

A comprehensive approach to ECDSA

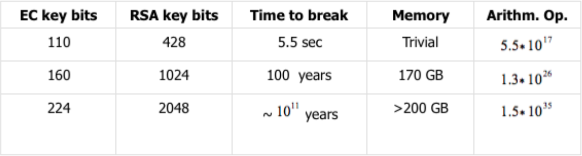
Francesco Leccese, Francesco Peverini

Abstract

This paper is about the mathematical patterns that are used on the digital signature algorithm on elliptic curves ECDSA (Elliptic Curves Digital Signature Scheme). Elliptic curves were introduced in cryp-tography only a few years ago and they were immediately successful because they allow to use shorter keys, hence a lower computational complexity, compared to other cryptographic algorithms. Based on that, ECDSA quickly became the most used Digital Signature algorithm, replacing DSA.

1. Introduction

Ever since ancient times, the pursuit of secrecy of information always played a primary role in human history. Simultaneously, a whole branch of mathematics developed to optimize the encryption and decryption processes of a message: Cryptography. In the military, cryptography has always been crucial: an important example is Julius Caesar, who already used rudimentary cryptographic tools (Caesar cypher) to give orders to his lieutenants on the battle ground; another considerable example is World War II, and the key role played by the cypher machine ENIGMA for Hitler and Germany. With the advent of the digital era, cryptography played a more important role than it did before, because in the modern world whatever is on the net is encrypted (and also signed) with algorithms that use algebraic-geometric concepts. In an increasingly faster and interconnected world, Digital Signature is of primary importance because it allows to uniquely identify and verify the author of a message. One of the current digital signatures standards is ECDSA (Elliptic Curves Digital Signature Algorithm), which exploits the Elliptic Curves over Finite Fields theory. The strength of ECDSA is given by the short-term intractability ECDLP (Elliptic Curves Discrete Logarithm Problem). Moreover, Elliptic curves cryptography offers significant benefits, such as the length of keys and memory costs, maintaining the same security level of other algorithms (for example RSA), as shown in the following table:



EC key bits	RSA key bits	Time to break	Memory	Arithm. Op.
110	428	5.5 sec	Trivial	$5.5 * 10^{17}$
160	1024	100 years	170 GB	$1.3 * 10^{26}$
224	2048	$\sim 10^{11}$ years	>200 GB	$1.5 * 10^{35}$

Table 1: Times to break with Sunway TaihuLight super-computer.

2. Elliptic Curves

In maths, elliptic curves are a specific kind of two variable plane curves. Given $a, b \in \mathrm{R}^2$ such that $4a^3 + 27b^2$ different from 0, we define as non-singular elliptic curve the set . $E=\{(x, y) \in \mathrm{R}^2\ t.c. y^2=x^3 + ax + b\}$
Given an elliptic curve E, it is possible to define an operation on its points, denoted as " + ": given P ,Q points of E, where P = (x_1, y_1) and Q = (x_2, y_2), we define P + Q as follows:

5. We draw the line r between P and Q.

6. We find T = (x3, y3) as the intersection point between the line r and the curve E.

7. Finally we obtain R = (x3,-y3) = P + Q as the symmetrical point of T through X.

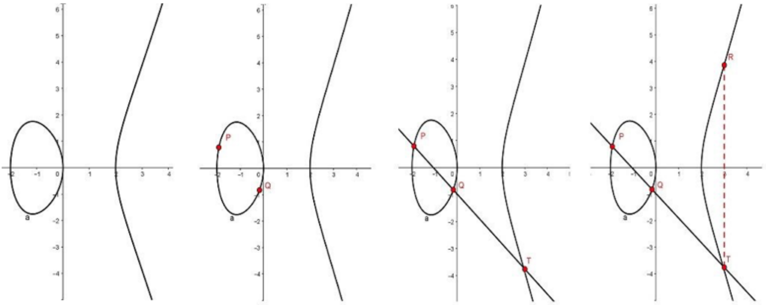


Figure 1: Graphical representation of P + O

The previous operation has some important properties, for example it is closed on E (i.e. if P , Q are two random points of E, then P + Q belongs to E) and is commutative (P + Q = Q + P). It is also possible to show that (E,+) is an abelian group, i.e. E is a set with a closed operation on itself that is also commutative, associative and has the reciprocal of every element.

3. Fq and the modulus calculus

Elliptic curves over finite fields, denoted with Fq, are used in ECDSA. In these particular fields the common operations as sum, multiplication and exponentiation are performed; however, the arithmetic steps are much more different from those we usually use. For example, let us consider the finite field modulo 17 $\mathrm{F}_q = \mathrm{Z}_{17} = \{0,1,...,16\}$, where every number is represented as the remainder of the division for 17 (e.g. 35 is represented as 1, because 35 = 2*17 + 1); assume that we want to calculate 10 + 28, 10*28 and 4^3. In the real field, we obtain 10 + 28 = 38, 10*28 = 280 and 4^3 = 64; instead, in Z17 we obtain :

$$10 + 28 = 38 = 2*17 + 4 = 4 \bmod 17;$$

$$10 *28 = 280 = 16*17 + 8 = 8 \bmod 17;$$

and

$$4^3 = 64 = 3*17 + 13 = 13 \bmod 17.$$

It is clearly more difficult to invert one of these operations in finite fields rather than in the real field (just consider solving the equation 7*x = 11 mod17). This aspect is basic in the ECDSA, and generally almost in the whole modern cryptography, because most algorithms base their security on the difficulty to invert modulus operation in short-time.

4. Elliptic Curves DSA Algorithm

We shall now see ECDSA in detail: let p be a big prime number ($p \approx 2^{256}$), E an elliptic curve over the finite field Fp, A a point belonging to E of order q (qA = A) (so as DLP is intractable in short time), and let $K = \{(p, q, E, A, m, B)\ t.c.\ B = mA\}$ be the set of all the possible keys. Finally, let us suppose we have a message M written in binary code. The public key will be the vector (p, q, E, A, B), while m will be the private key. We now have to choose a random (secret) number k (0 < k < q-1): we can then calculate the digital signature of M, $sig_K\,(M, k)=(r, s)$, as follows: first of all we calculate

$$kA=A+...+A=(u, v)$$

and we define

$$r = u \bmod q\ ,$$
$$s= k^{(-1)}(SHA\text{-}1(M)+mr) \bmod q$$

The verification process receives (M; (r; s)) as input, and we obtain

$$w = s^{-1} \bmod q\ ,$$
$$i = w\ SHA\text{-}1\ (M) \bmod q\ ,$$

$$j= wr \bmod q\ ,$$

$$(u, v) = iA + jB$$

finally, the digital signature will be accepted if and only if

$$u \ mod \ q = r$$

The first idea that might come to mind when considering an attack the ECDSA is to somehow obtain k, because if an attacker knows k and knows that:

$$s = k^{(-1)} (SHA\text{-}1 \ (M) + mr) \ mod \ q$$

then he can get m in a few simple steps, which require the calculation of the discrete logarithm logAB: given E an elliptic curve on the finite field F_q and A, B two points belonging to E, the attacker would have to find $k \in F$ such that

kA = B,

We shall now see an example of the difficulty of DLP: suppose we want to solve the equation

$$4^k = 13mod \ 17$$

one of the fastest algorithms to find k is brute force, i.e. to substitute to k all the possible numbers belonging to Z_{17}. In this example, an attacker would need 17 tries to obtain the solution k = 3, but what if he uses brute force on a ECDSA key? On average, he would need 2^{80} operations and more or less 30 million years to run these. Some algorithms which improve the search of discrete logarithm (Pollard's Rho, Shank's Algorithm and others) already exist, although they need too much time for current computers. Despite the strength of ECDLP, there are some compulsory precautions to avoid an attack of the system by a hacker: for example, the value of q should not be factorizable in small prime numbers (Pohlig-Hellman Attack), and a same value for k should never be used in two different digital signatures; otherwise, it becomes easy to get k (as well as the private key m), just as ahappened to Sony in 2010.

7. APPENDIX B

List of aconyms used in this paper

AES	Advenced Encryption Standard
AISP	Account Information Services Provider
API	Application Programming Interface
ASPSP	Account Servicing Payments Service Provider
DLP	Discrete Logarithm Problem
DSA	Digital Signature Algorithm
EBA	European Banking Authority
ECB	European Central Bank
ECDLP	Elliptic Curves Discrete Signature Algorithm
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
IBAN	International Bank Account Number
ID	Identity Document
ISO	International Organization for Standardization
ISTAT	Istituto Nazionale di Statistica (Central Statistics Institute)
IV	Initiation Value
PKI	Public Key Infrastructure
PISP	Payment Initiation Services Provider
PoW	Proof-of-Work
PSD	Payment Service Directive
PSD2	Revised Payment Services Provider
PSU	Payment Service User
RSA	Rivest Shamir Aldeman
RTS	Regulatory Technical Standards
SCA	Strong Customer Authentication
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
REST	REpresentational State Transfer
SEPA	Single Euro Payments Area
SMTP	Simple Mail Transfer Protocol
TCP-IP	Transmission Control Protocol - Internet Protocol
TPP	Third-Part Provider
UDDI	Universal Description Discovery and Integration
WSDL	Web Service Definition Language
XML	eXtensible Stylesheet Language
XMPP	eXtensible Messaging and Presence Protocol

Bio

Francesco Corona is a lecturer and director of the Hacker and Engineering Security Master at Link Campus University Rome, formerly a lecturer at the Management Engineering Faculty of Rome2 "Tor Vergata" University (Business Engineering Department) He has carried out intense research activities in the MIUR (Ministry of University and Scientific Research) field and professional activity in the cyber security sector on behalf of leading national and international players. In 2013 professor Corona obtained the prestigious National Innovation Award and the special ICT Confindustria Award. f.corona@unilink.it.

Francesco Leccese, graduated in Mathematics. He received a three-year degree on 25Th February 2015, at Università degli Studi di Perugia. He's taking a master degree in Mathematics for IT Security with a cryptographic-based thesis focused on Elliptic Curves applied on Blockchain. Currently he works as apprentice at GCSEC, dealing with a project about possible ways of implementation of Blockchain and new cryptographic technologies in PSD2 area.

Francesco Peverini, graduated in Mathematics. He received a master degree on 28Th April 2017, at Università degli Studi di Perugia. His thesis work was focused on cryptography. In particular, he worked on the creation of an innovative Secret Sharing Scheme compared to those already known. From July 2017 he works as apprentice at GCSEC, dealing with a project about possible ways of implementation of Blockchain and new cryptographic technologies in PSD2 area.

Luca Faramondi received the Laurea degree in Computer Science and Automation (2013) and in 2014 he was finalist for the CIPRNet Young Critis Award. He was honored in 2016 for the best master degree in the field of critical infrastructure (AIIC). He received the PhD degree on Computer Science and Automation (2017) from the University Roma Tre of Rome. He is currently PostDoc Fellow at Complex Systems & Security Laboratory at the University Campus Bio-Medico of Rome. He is involved in several national and European projects about the Critical Infrastructure and Indoor Localization. His research interests include the identification of network vulnerabilities, cyber physical systems, and optimization at large. He is part of the technical comitee of conferences and books, and reviewer for international journals and conferences.

Global
Cyber Security
Center

GCSEC - Global Cyber Security Center – Viale Europa, 175 00144 Rome, Italy – www.gcsec.org