

الجمهورية الجزائرية الديمقراطية الشعبية  
The People's Democratic Republic of Algeria  
وزارة التعليم العالي والبحث العلمي  
Ministry of Higher Education and Scientific Research



جامعة ابن خلدون تيارت  
University of Ibn Khaldoun TIARET  
كلية الرياضيات والإعلام الآلي  
Faculty of Mathematics and Computer Science  
قسم الإعلام الآلي  
Department of Computer Science



## Master Thesis in Computer Science Networks and Telecommunications

3D-Secure electronic payment architecture and adaptive  
authentication for Ecommerce

**Realized by:**

- Boucherit Lahecen
- Guelfout Nasr Eddine

**Supervised by:**

- Pr. Dahmani Youcef

Defense on: Oct 20, 2020

**Examiners:**

**President:** Dr. Nassane samir  
**Assessor 1:** Dr. Bakkar khaled

**Academic Year: 2019-2020**



## Acknowledgement

In the name of Allah the most Merciful and Beneficent

First and foremost,

# ALLAH

All Praise is to **ALLAH** S.W.T, the Almighty, the

greatest of all, on whom ultimately we depend for sustenance and guidance. We would like to thank Almighty Allah for giving us opportunity, determination and strength to do our study. His continuous grace and mercy was with us throughout our life and ever more during the tenure of our study.

To our **Parents**, we may not always say how much we love both of you but we can say we have a big place in our hearts for you two. Thank you, moms and dads, for raising us so perfectly. **ALLAH** has blessed us in so many ways, but the biggest of them all is our parents. They deserve the best from us always. Thank you to all the parents in the world!.

Now, we would like to thank and express our deep and sincere gratitude to our supervisor **Pr. Dahmani Youcef**, Dean of Mathematics and Computer Sciences Faculty, University of Ibn Khaldoun **TIARET** for his continuous support, guidance and encouragement. We appreciate all his contributions of time, support and ideas.

We would also like to kindly thank our examiners **Dr. Nassane Samir** and **Dr. Bakkar khaled**, for their judgments of this work and for their valuable notes and suggestions.

We owe everything to our **families** who encouraged and helped us at every stage of our personal, professional and academic life. We love you all. We would like to thank our colleagues and friends.

## ملخص

في أيامنا الحالية، عدد المستهلكين الذين يرغبون في الخروج للتسوق في تناقص مستمر، لأنه يمكن لهؤلاء المستهلكين التسوق عبر الإنترنت دون مغادرة منازلهم. هذا النوع من التسوق يسمى التجارة الإلكترونية. أصبحت التجارة الإلكترونية خدمة مفيدة للمستهلكين، وهي أيضاً عنصر مهم في الأنشطة اليومية للتجار. لأنها تسمح لهم بتقليل تكاليف العملاء والمؤسسات، وتسمح لهم بتوفير الكثير من الوقت، كما يمكن أن تقدم لهم الكثير من الفوائد الأخرى. ومع ذلك، يوجد عوائق للتجارة الإلكترونية من بينها الافتقار إلى الأمان في المعاملات الإلكترونية والسهولة التي يمكن بها انتهاك خصوصية الاتصالات عبر الإنترنت.

تعتبر مواقع المتاجر الإلكترونية هدفاً سهلاً للقراصنة. ويتمثل التحدي في الدمج الناجح للتدابير والآليات الأمنية الفعالة لحماية الأعمال من التعرض لخطر القرصنة. حيث أننا متفقون أن الأمان مهم وجزء من حياتنا اليومية.

هناك الكثير من استراتيجيات وآليات أمن التجارة الإلكترونية، أحدها بروتوكول 3D Secure. وهو بروتوكول مراسلة تم تطويره بواسطة EMVCo لتمكين المستهلكين من التعريف بأنفسهم مع جهة إصدار البطاقة الخاصة بهم عند إجراء عمليات الشراء عن طريق التجارة الإلكترونية والبطاقة غير الموجودة (CNP). تساعد طبقة الأمان الإضافية في منع معاملات CNP غير المصرح بها وتحمي التاجر من تعرضه للاحتيال في حالة غياب البطاقة. ويتكون هذا البروتوكول من ثلاث مجالات وهي مجال التاجر/المشتري ومجال الزبون/المصدر ومجال الوسيط (التشغيل البيئي).

تهدف الوثيقة الحالية إلى وصف كيفية تجسيد البروتوكول 3D Secure وفقاً لمعيار EMV وتكيفية مع سوق التجارة الإلكترونية في الجزائر، وكذلك كيفية إجراء المصادقة المزدوجة باستخدام الرسائل القصيرة والبريد الإلكتروني عبر كلمة المرور لمرة واحدة (OTP)، أو آليات أخرى. ومع ذلك، يجب القيام بمزيد من العمل لتعزيز الأمان ضد نمو عمليات الاحتيال والهجمات على المعاملات التجارية عبر الإنترنت. ونأمل أن نتمكن في المستقبل القريب من التعرف الصحيح والفعال على حامل البطاقة دون المصادقة الثنائية باستخدام الذكاء الاصطناعي (رؤية الحاسوب) والتعرف على الوجوه، ويكون هذا بإلغاء استعمال OTP وبدون تدخل المستخدم.

**الكلمات المفتاحية:** البروتوكول 3D Secure، التجارة الإلكترونية، نظام الدفع الإلكتروني، بوابة الدفع، أمان الدفع، المصادقة، سرقة معلومات البطاقة، EMV.

# Abstract

In our present days, fewer and fewer consumers want to travel to shop, through the Internet these consumers can do all their shopping without leaving their homes. This type of shopping is called e-commerce. E-Commerce is becoming a useful service for consumers, but it is also an important component in the daily activities of merchants. It allows them to reduce the costs of the customers and enterprises, and it allows them to save a lot of time, it can also offer them a lot of benefits. However, the lack of security in web-based transactions and the ease with which the privacy of online communications can be violated are the main stumbling blocks of e-commerce.

Merchant online shop websites provide an easy target for attackers. The challenge is to successfully integrate effective security measures and mechanisms to protect the business from being compromised by attackers. We know that effective security is important and has become part of our daily life.

There are a lot of E-commerce security strategies and mechanisms, one of them is 3D Secure protocol. Which is a messaging protocol developed by EMVCo to enable consumers to authenticate themselves with their card issuer when making card-not-present (CNP) e-commerce purchases. The additional security layer helps prevent unauthorized CNP transactions and protects the merchant from CNP exposure to fraud. The three domains consist of the merchant/acquirer domain, customer/issuer domain, and the interoperability domain.

The present document aims to describe how to implement the 3D secure protocol according to the EMV standard and adapt it for the ecommerce market in Algeria, and also how to make the double authentication using SMS and EMAIL via one-time password (OTP), or other mechanisms. However, further work needs to be done to enhance the security against the growth of online transactions frauds and attacks. We hope that in the near future we can identify the cardholder without the duplicate authentication using AI (computer vision) and face recognition and without OTP and without any user interaction.

**Keywords:** 3D Secure protocol, E-commerce, Electronic Payment System, Payment Gateway, Payment Security, Authentication, Card fraud, EMV.

# Contents

Acknowledgement .....	i
ملخص .....	ii
Abstract .....	iii
Contents .....	iv
Figures.....	vii
Tables.....	viii
Abbreviations.....	ix
Definitions and terminology .....	x
Introduction.....	1
Chapter 01: E-Payment .....	3
1. Introduction .....	4
2. History of E-commerce .....	4
3. Types of Ecommerce .....	5
• Business-to-Consumer (B2C).....	5
• Business-to-Business (B2B).....	5
• Consumer-to-Consumer (C2C). .....	5
4. Characteristics of e-commerce technologies .....	5
4.1. Ease of automated processing .....	5
4.2. Immediacy of result.....	5
4.3. Openness and accessibility.....	5
4.4. Loss of collateral information .....	5
4.5. Globalization .....	6
4.6. New business models .....	6
5. Advantages and disadvantages .....	6
6. Conclusion .....	7
Chapter 02: E-Payment Security.....	8
1. Introduction .....	9
2. Security Services .....	9
2.1. Access Control (AC).....	9
2.2. Authentication .....	9
2.3. Authorization.....	9
2.4. Availability.....	9
2.5. Data Confidentiality .....	10
2.6. Data Integrity .....	10
3. Card payment security models .....	10
3.1. No security model .....	10
3.2. Through third-party brokers paid model .....	10
3.3. Simple encrypted payment system model.....	11
3.4. SET (security electronic transaction) model.....	12
4. Security protocols .....	12
4.1. Most popular protocols .....	12
4.2. Typical transaction process .....	14
5. The 3D Secure System .....	15
5.1. Architecture:.....	15

5.2. Typical transaction process of 3D Secure.....	17
5.3. Four-party of 3D Secure system .....	21
6. Alternatives.....	23
06.1. PayPal.....	23
6.2. iDEAL .....	23
7. Conclusion .....	23
Chapter 03: Design and analysis of 3D Secure.....	24
1. Introduction:.....	25
2. 3D Secure protocol components.....	25
2.1. Acquirer Domain.....	26
2.2. Interoperability Domain .....	26
2.3. Issuer Domain .....	26
3. Messages used by 3D Secure protocol [9] .....	27
3.1. Authentication Request Message (AReq) .....	27
3.2. Authentication Response Message (ARes) .....	27
3.3. Challenge Request Message (CReq).....	27
3.4. Challenge Response Message (CRes).....	27
3.5. Results Request Message (RReq) .....	27
3.6. Results Response Message (RRes) .....	27
3.7. Preparation Request Message (PReq) .....	27
3.8. Preparation Response Message (PRes) .....	28
3.9. Error Message .....	28
4. Global Activity diagram .....	28
5. The 3D Secure process under online shopping .....	29
Step 01: Make online shopping by cardholder.....	29
Step 02: 3D Secure initiation .....	30
Step 03: Submit information card to Payment Gateway.....	31
Step 04: Card information verification (1 <sup>st</sup> Auth) .....	32
Step 05: Cardholder verification (2 <sup>nd</sup> Auth).....	33
Step 06: Make transaction.....	34
Step 07: Ending process .....	35
6. Pseudocode .....	36
6.1. Frontend pseudocode: .....	36
6.2. Backend pseudocodes .....	36
7. Conclusion .....	39
Chapter 04: Implementation & Results .....	40
1. Introduction: .....	41
2. Environment description .....	41
2.1. Hardware .....	41
2.2. Software .....	42
3. Web Application developed .....	45
3.1. Banks web application: .....	45
3.2. Ecommerce web app: .....	48
4. Source codes added to implement 3D Secure .....	54
5. Results .....	54
5.1. Table of parameters.....	54
5.2. Online Shopping Steps using 3D Architecture .....	56
Step 01: Make online shopping by cardholder.....	56

---

Step 02: Submit card information to Payment Gateway .....	59
Step 03: Send code .....	59
Step 04: verification email .....	60
Step 05: type code .....	61
Step 06: Make transaction .....	61
Step 07: payment receipt .....	63
Step 08: Send payment receipt via email .....	64
6. Conclusion .....	65
Conclusion .....	66
Bibliography .....	67
Appendix .....	68
Appendix A: Message Format .....	68
A.1. AReq Message Data Elements .....	68
A.2. ARes Message Data Elements .....	69
A.3. CReq Message Data Elements .....	70
A.4. CRes Message Data Elements .....	70
A.5. Final CRes Message Data Elements .....	71
A.6. PReq Message Data Elements .....	71
A.7. PRes Message Data Elements .....	71
A.8. RReq Message Data Elements .....	71
A.9. RRes Message Data Elements .....	72
A.10. Error Messages Data Elements .....	72
Appendix B: 3D Secure Implementation (Source codes) .....	73
B.1. Payment Gateway .....	73
B.2. MPI (Merchant Plug-In) .....	75
B.3. Issuer requirements .....	78
B.4. Acquirer requirements .....	82
B.5. Interoperability requirements .....	84



# Figures

Figure 1: No security model. [6].....	10
Figure 2: Through third-party brokers paid model. [6] .....	11
Figure 3: Simple encrypted payment system model. [6] .....	11
Figure 4: SET (security electronic transaction) Model. [6].....	12
Figure 5: Typical transaction process. [7] .....	14
Figure 6: 3D Secure Architectural Overview. [8].....	16
Figure 7: Verify Enrolment. [8] .....	17
Figure 8: Cardholder Authentication. [8] .....	18
Figure 9: Merchant Web Page With 3D Secure IFRAME. [8].....	19
Figure 10: 3D Secure Authentication Using A One Time Password Via SMS. [8].....	20
Figure 11: Four party system. [8] .....	21
Figure 12: 3D Secure Domains and Components [9].....	26
Figure 13: Global Activity diagram.....	28
Figure 14: Sequence diagram of SD Secure process. [9] .....	29
Figure 15: First step to do an online shopping.....	30
Figure 16: Second step, checkout and choose payment method. ....	31
Figure 17: The third step, fill information card. ....	32
Figure 18: The fourth step, 1 <sup>st</sup> authentication. ....	33
Figure 19: The fifth step, 2 <sup>nd</sup> authentication. ....	34
Figure 20: Bank to bank transaction. ....	35
Figure 21: Global architecture of 3D Secure environment.....	42
Figure 22: Use cases of Banking web app.....	45
Figure 23: Sequence diagram of Banking web app. ....	46
Figure 24: Database diagram of Banking web app.....	47
Figure 25: Guest use cases of ecommerce web app.....	48
Figure 26: Customer use cases of ecommerce web app .....	49
Figure 27: Admin use cases of e-commerce web app. ....	49
Figure 28: Guest sequence diagram of ecommerce web app.....	50
Figure 29: Customer sequence diagram of ecommerce web app .....	50
Figure 30: Admin sequence diagram of ecommerce web app.....	51
Figure 31: Database diagram of ecommerce web app.....	52
Figure 32: Issuer accounts. ....	55
Figure 33: Acquirer accounts.....	55
Figure 34: Hanouti web app home page. ....	56
Figure 35: Hanouti login page .....	56
Figure 36: Hanouti add products to cart. ....	57
Figure 37: Hanouti checkout page. ....	57
Figure 38: Hanouti, choose the payment method. ....	58
Figure 39: PGW-PFE3DS, information card page. ....	59
Figure 40: PGW-PFE3DS, send verification code. ....	59
Figure 41: Issuer send verification email.....	60
Figure 42: PGW-PFE3DS, Enter the code.....	61
Figure 43: Issuer transaction.....	61
Figure 44: Issuer account balance.....	62

Figure 45: Acquirer transaction .....	62
Figure 46: Acquirer account. ....	63
Figure 47: Hanouti payment receipt. ....	63
Figure 48: Hanouti, send payment receipt via email. ....	64

## Tables

Table 1: Definitions and terminology. ....	xi
Table 2: History of Ecommerce [2]. ....	4
Table 3: Advantages of E-Commerce. ....	6
Table 4: Disadvantages of E-Commerce. ....	7
Table 5: Comparison of SSL and SET protocols. [7] .....	13
Table 6: Characteristics of physical and virtual machines. ....	41
Table 7: Features list of Banking web app. ....	48
Table 8: Features list of ecommerce web app. ....	53
Table 9: Customers parameters. ....	54
Table 10: Accounts parameters. ....	54
Table 11: Cards parameters. ....	55
Table 12: AReq Data Elements. ....	69
Table 13: ARes Data Elements. ....	70
Table 14: CReq Data Elements. ....	70
Table 15: CRes Data Elements. ....	71
Table 16: Final CRes Data Elements. ....	71
Table 17: PReq Data Elements. ....	71
Table 18: PRes Data Elements. ....	71
Table 19: RReq Data Elements. ....	72
Table 20: RRes Data Elements. ....	72
Table 21: Error Message Data Elements. ....	72

## **Abbreviations**

The abbreviations listed below are used in this specification.

<b>3DS:</b>	Three Domain Secure
<b>3DS SDK:</b>	Three Domain Secure Software Development Kit
<b>ACS:</b>	Access Control Server
<b>AReq:</b>	Authentication Request
<b>ARes:</b>	Authentication Response
<b>BIN:</b>	Bank Identification Number
<b>CA:</b>	Certificate Authority
<b>CA DS:</b>	Certificate Authority Directory Server
<b>CReq:</b>	Challenge Request
<b>CRes:</b>	Challenge Response
<b>DS:</b>	Directory Server
<b>JSON:</b>	JavaScript Object Notation
<b>MAC:</b>	Message Authentication Code
<b>PA:</b>	Payment Authentication
<b>OTP:</b>	One-time Passcode
<b>PReq:</b>	Preparation Request Message
<b>PRes:</b>	Preparation Response Message
<b>RReq:</b>	Results Request Message
<b>RRes:</b>	Results Response Message
<b>RSA:</b>	Rivest-Shamir-Adleman
<b>SDK:</b>	Software Development Kit
<b>TLS:</b>	Transport Layer Security
<b>URL:</b>	Uniform Resource Locator
<b>UUID:</b>	Universally Unique Identifier

# Definitions and terminology

The following terms are used in this specification:

Term	Definition
3DS Client	The consumer-facing component allowing consumer interaction with the 3DS Requestor for initiation of the EMV 3D Secure protocol.
3DS Integrator	An EMV 3D Secure participant that facilitates and integrates the 3DS Requestor Environment, and optionally facilitates integration between the Merchant and the Acquirer.
3DS Method	A scripting call provided by the 3DS Integrator that is placed on the 3DS Requestor website. Optionally used to obtain additional browser information to facilitate risk-based decisioning.
3DS Requestor	The initiator of the EMV 3D Secure Authentication Request. For example, this may be a merchant or a digital wallet requesting authentication within a purchase flow.
3DS Requestor App	An App on a Consumer Device that can process a 3D Secure transaction through the use of a 3DS SDK. The 3DS Requestor App is enabled through integration with the 3DS SDK.
3DS Requestor Environment	The 3DS Requestor-controlled components (3DS Requestor App, 3DS SDK, and 3DS Server) are typically facilitated by the 3DS Integrator. Implementation of the 3DS Requestor Environment will vary as defined by the 3DS Integrator.
3DS Requestor Initiated (3RI)	3D Secure transaction initiated by the 3DS Requestor for the purposes of confirming that an account is still valid or for Cardholder authentication. The first main use case being recurrent transactions (TV subscriptions, utility bill payments, etc.) where the merchant wants to perform a payment transaction to receive authentication data for each bill or a non-payment transaction to verify that a subscription user still has a valid form of payment. The second main use case is when the 3DS Requestor requests Decoupled Authentication as a method to authenticate the Cardholder.
3DS Requestor Website	Component that provides the website that requests Cardholder credentials (whether on file or entered by Cardholder).
3DS SDK	3D Secure Software Development Kit (SDK). A component that is incorporated into the 3DS Requestor App. The 3DS SDK performs functions related to 3D Secure on behalf of the 3DS Server.
3DS Server	Refers to the 3DS Integrator's server or systems that handle online transactions and facilitates communication between the 3DS Requestor and the DS.
3D Secure (3DS)	An e-commerce authentication protocol that enables the secure processing of payment, non-payment and account confirmation card transactions.
Access Control Server (ACS)	A component that operates in the Issuer Domain, that verifies whether authentication is available for a card number and device type, and authenticates specific Cardholders.
Acquirer	A financial institution that establishes a contractual service relationship with a Merchant for the purpose of accepting payment cards. In the context of 3D Secure, in addition to the traditional role of receiving and sending authorization and settlement messages (enters transaction into interchange), the Acquirer also determines whether a Merchant is eligible to support the Merchant's participation in 3D Secure.
Authentication Request (AReq) Message	An EMV 3D Secure message sent by the 3DS Server via the DS to the ACS to initiate the authentication process.
Authentication Response (ARes) Message	An EMV 3D Secure message returned by the ACS via the DS in response to an Authentication Request message.
Certificate	An electronic document that contains the public key of the certificate holder and which is attested to by a Certificate Authority (CA) and rendered not forgeable by cryptographic technology (signing with the private key of the CA).
Certificate Authority	A trusted party that issues and revokes certificates. Refer also to DS Certificate Authority.
Challenge	The process where the ACS is in communication with the 3DS Client to obtain additional information through Cardholder interaction.
Challenge Flow	A 3D Secure flow that involves Cardholder interaction as defined in Section 2.5.2.
Challenge Request (CReq) Message	An EMV 3D Secure message sent by the 3DS SDK or 3DS Server where additional information is sent from the Cardholder to the ACS to support the authentication process.

Challenge Response (CRes)	The ACS response to the CReq message. It can indicate the result of the Cardholder authentication or, in the case of an App-based model, also signal that further Cardholder interaction is required to complete the authentication.
Digital signature	An asymmetric cryptographic method whereby the recipient of the data can prove the origin and integrity of data, thereby protecting the sender of the data and the recipient against modification or forgery by third parties and the sender against forgery by the recipient.
Directory Server (DS)	A server component operated in the Interoperability Domain; it performs a number of functions that include: authenticating the 3DS Server, routing messages between the 3DS Server and the ACS, and validating the 3DS Server, the 3DS SDK, and the 3DS Requestor.
Directory Server Certificate Authority (DS CA)	A component that operates in the Interoperability Domain; generates and distributes selected digital certificates to components participating in 3D Secure. Typically, the Payment System to which the DS is connected operates the CA.
EMV	A term referring to EMVCo's specifications for global interoperability and acceptance of secure payment transactions and/or products and services complying with such specifications.
EMVCo	EMVCo, LLC, a limited liability company incorporated in Delaware, USA.
Ends 3D Secure Processing	In the 3D Secure processing flow, this indicates that no further processing as defined by this specification will be performed. Per merchant preferences, an authorisation transaction may still be performed although it will happen without a successful 3D Secure authentication outcome.
Issuer	A financial institution that issues payment cards, contracts with Cardholders to provide card services, determines eligibility of Cardholders to participate in 3D Secure, and identifies for the Directory Server card number ranges eligible to participate in 3D Secure.
JavaScript Object Notation (JSON)	An open standard format that uses human-readable text to transmit data objects consisting of attribute-value pairs. It is typically used to transmit data between a server and web application. Refer to Table 1.1 for RFC references.
Key	In cryptography, the value needed to encrypt and/or decrypt a value.
Merchant	Entity that contracts with an Acquirer to accept payment cards. Manages the online shopping experience with the Cardholder, obtains card number, and then transfers control to the 3DS Server, which conducts payment authentication.
One-Time Passcode (OTP)	A passcode that is valid for only one login session or transaction, on a computer system or other digital device.
Payment System	A Payment System defines the operating rules and conditions, and the requirements for card issuance and Merchant acceptance.
Preparation Request (PReq) Message	3D Secure message sent from the 3DS Server to the DS to request the ACS and DS Protocol Version(s) that correspond to the DS card ranges as well as an optional 3DS Method URL to update the 3DS Server's internal storage information.
Preparation Response (PRes) Message	Response to the PReq message that contains the DS Card Ranges, active Protocol Versions for the ACS and DS and 3DS Method URL so that updates can be made to the 3DS Server's internal storage.
Protocol Version	Refers to the version of the EMV 3D Secure specification that the component supports. The protocol version for this specification is 2.1.0.
Results Request (RReq) Message	Message sent by the ACS via the DS to transmit the results of the authentication transaction to the 3DS Server.
Results Response (RRes) Message	Message sent by the 3DS Server to the ACS via the DS to acknowledge receipt of the Results Request message.
Transport Layer Security (TLS)	A cryptographic protocol developed by the IETF (Internet Engineering Task Force) to confidentially transmit information over open networks, such as the Internet. Refer to Table 1.1 for RFC references.
Uniform Resource Locator (URL)	Address scheme for pages on the World Wide Web usually in the format <a href="http://www.example.com">http://www.example.com</a> or <a href="https://www.example.com">https://www.example.com</a> .
Universally Unique Identifier (UUID)	Identifier standard used in software construction. In its canonical form, a UUID is represented by 32 lowercase hexadecimal digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 alphanumeric characters and four hyphens). Refer to Table 1.1 for RFC references.

**Table 1: Definitions and terminology.**

# Introduction

Until these days, for many hundreds of years or since the existence of the human being, people were shopping and sometimes traveled to do this to improve their lifestyles. Nowadays, fewer and fewer consumers want to travel to shop, through the Internet these consumers can do all their shopping without leaving their homes. This type of shopping is called e-commerce.

E-Commerce is becoming a useful service for consumers, but it is also an important component in the daily activities of merchants. It allows them to contact their customers and suppliers, advertise and even organize efficiently the invoicing and the distribution of their products and services. In addition, it reduces the operating and support costs of the business. However, the lack of security in web-based transactions and the ease with which the privacy of online communications can be violated are the main stumbling blocks of e-commerce.

Merchant online shop websites provide an easy target for attackers because they typically have limited funds and do not have dedicated personnel to monitor, update and defend their systems. The attacks on businesses continue to rise each year. The challenge is to successfully integrate effective security measures and mechanisms to protect the business from being compromised by attackers. Effective security is important for the continuity of business, trust of clients, and compliance with industry-specific laws and regulations. One breach in security can cost a business a lot of money, even shut it down.

There are a lot of E-commerce security strategies and mechanisms, one of them is 3D Secure protocol. Which is a messaging protocol developed by EMVCo to enable consumers to authenticate themselves with their card issuer when making card-not-present (CNP) e-commerce purchases. The additional security layer helps prevent unauthorized CNP transactions and protects the merchant from CNP exposure to fraud. The three domains consist of the merchant/acquirer domain, customer/issuer domain, and the interoperability domain.

The present document aims to describe how to implement the 3D secure protocol according to the EMV standard and adapt it for the ecommerce market in Algeria, and also how to make the double authentication using SMS and email via one-time password (OTP), or other mechanisms.

This document is divided into three main sections. The first section gives an overview of ecommerce and E-payment security (the state of the art), this section is organized as follows: the first chapter begins by a history of Ecommerce, following this by its types, its characteristics, and finally its pros and cons. In the second chapter we will show the security services or requirements, the different security models for E-payment using cards, the most famous protocols used in online shopping, and a detailed study of 3D Secure System and a summary of two alternatives that implement 3D Secure scheme.

The second section examines 3D Secure protocol components and messages, a general activity diagram of the protocol is outlined in this section, a detailed and explained steps of this protocol, and finally we will show the 3D Secure pseudocode.

In the last section, we have described our environment including hardware and software tools used, and we have analyzed and presented the different diagrams used to develop the three web applications for banks, payment gateway, and merchant online shop. After that, we will present source code developed to implement our protocol, and by making a real demonstration, we have obtained comprehensive results proving our implementation.

# **Chapter 01: E-Payment**



## 1. Introduction

Electronic commerce is a powerful concept and process that has fundamentally changed the current of human life. Electronic commerce is one of the main criteria of revolution of Information Technology and communication in the field of economy. This style of trading due to the enormous benefits for human has spread rapidly. Certainly, can be claimed that electronic commerce is canceled many of the limitations of traditional business. For example, form and appearance of traditional business has fundamentally changed. These changes are basis for any decision in the economy. Existence of virtual markets, passages and stores that have not occupy any physical space, allowing access and circulation in these markets for a moment and anywhere in the world without leaving home is possible. Select and order goods that are placed in virtual shop windows at unspecified parts of the world and also are advertising on virtual networks and payment is provided through electronic services, all of these options have been caused that electronic commerce is considered the miracle of our century. [1] In this chapter we will begin by a history of Ecommerce, following this by its types, its characteristics, and finally its advantages and disadvantages.

## 2. History of E-commerce

Year	Major Ecommerce Event
1969	The first major ecommerce company, CompuServe, is founded.
1979	Michael Aldrich invents electronic shopping.
1982	Boston Computer Exchange launches as one of the first ecommerce platforms.
1992	Book Stacks Unlimited launches as one of the first online marketplaces for books.
1994	Netscape launches Netscape Navigator, an early web browser, making it easier for users to browse online.
1995	Amazon and eBay launch.
1998	PayPal launches as an online payment system.
1999	Alibaba.com launches.
2000	Google launches AdWords as an online search advertising tool.
2005	Amazon launches Amazon Prime with expedited, flat-fee shipping for members.
2005	Esty, an online marketplace for handmade and vintage goods launches.
2009	BigCommerce launches as an online storefront platform.
2009	Square, Inc. is founded.
2011	Google Wallet launches as an online payment system.
2011	Facebook launches sponsored stories as a form of early advertising.
2011	Stripe launches.
2014	Apple Pay launches as a form of mobile payment.
2014	Jet.com launches.
2017	Instagram shoppable posts are introduced.
2017	Cyber Monday sales exceed \$6.5B.

**Table 2: History of Ecommerce [2].**

### 3. Types of Ecommerce

Generally, there are many models of ecommerce that businesses can be categorized into, and the most important of them are [2]:

- **Business-to-Consumer (B2C).**

B2C ecommerce encompasses transactions made between a business and a consumer. This is one of the most widely used sales models in the ecommerce context. When you buy products from an online retailer, it is a business-to-consumer transaction.

- **Business-to-Business (B2B).**

B2B ecommerce relates to sales made between businesses, such as a manufacturer and a wholesaler or retailer. This type of ecommerce is not consumer-facing and happens only between business entities.

- **Consumer-to-Consumer (C2C).**

One of the earliest forms of ecommerce is the C2C ecommerce business model. This would include customer to customer selling relationships like those seen on eBay or Amazon, for example.

### 4. Characteristics of e-commerce technologies

In this title, we'll look at six specific factors that are generally present on well-designed ecommerce business.

#### 4.1. Ease of automated processing

A payer can now easily automate the generation and processing of multiple payments with minimal effort and cost. Previously, the dependency upon banks to handle most payments and the lack of a cheap, ubiquitous communications technology made automation of payment processes expensive and difficult to establish.

#### 4.2. Immediacy of result

Payment immediacy occurs because of automation and the ability of the intermediate systems and providers to process payments in real-time. In manual, paper-based systems there exists a time delay due to the requirement of human intervention in the process.

#### 4.3. Openness and accessibility

The availability of cheap computing and communications technology, and appropriate software enables small enterprises and individuals to access or provide a range of payment services that were previously only available to large organizations via dedicated networks or the transactional processing units of banks.

#### 4.4. Loss of collateral information

The new technology dispenses with, or alters, collateral information accompanying transactions. This information has traditionally been part of the transaction, and has been relied upon by the transacting parties to validate individual payments.

Collateral information can be defined as information:

- Which is not essential to the meaning and intent of a transaction
- Which is typically incidental to the nature of the communications channel over which the transaction is conducted; but nevertheless
- Provides useful contextual information for one or more of the parties to the transaction.

Collateral information can include many things ranging from tone of voice in a telephone call to the business cards and letterheads and apparent authority of the person with whom the firm is dealing. Since information is received only via a single channel (such as an electronic message) in electronic systems, new processes are needed to support and reinforce payments in the same way as manual systems.

#### 4.5. Globalization

Globalization, or the minimization of geographical factors in making payments, is an obvious aspect of the new payments systems. Its effect is upon areas such as size of the payments marketplace, uncertainty as to legal jurisdiction in the event of disputes, location and availability of transaction trails, and the ability of a payment scheme to rapidly adapt to regulatory regimes imposed by one country by moving to another.

#### 4.6. New business models

New business models are being developed to exploit the new payment technologies, in particular to address or take advantage of the disintermediation of customers from traditional payment providers such as banks. Disintermediation is where the technology enables a third party to intervene between the customer and the banking system, effectively transferring the customer's trusted relationship with the bank to the new party. [3]

### 5. Advantages and disadvantages

E-Commerce has many advantages. However, as we know it from every area of our life, there is "no free lunch". Of course, E-Commerce has some disadvantages (see tables 2 and 3). [4]

Advantages	
...for the customer	...for the provider
<ul style="list-style-type: none"> <li>• Flexible shopping hours (7/24h)</li> <li>• No waiting queues (if net is available and software appropriately designed)</li> <li>• Shopping at home (we don't have to leave our apartment, refuel our car or buy a subway ticket, look for a parking place, etc.)</li> <li>• Individual needs can be covered (if customization is offered)</li> <li>• Global offers, more competition, pressure on prices</li> </ul>	<ul style="list-style-type: none"> <li>• Better customer service can be offered</li> <li>• Fast communication with customer</li> <li>• New customer potential through global visibility</li> <li>• No (traditional) intermediaries, who take away margins</li> </ul>

**Table 3: Advantages of E-Commerce.**

Disadvantages	
...for the customer	...for the provider
<ul style="list-style-type: none"> <li>• Security risks:               <ol style="list-style-type: none"> <li>1. Data theft (e.g. stealing account or credit card numbers)</li> <li>2. Identity theft (acting under our name or user identity)</li> <li>3. Abuse (e.g. third person orders goods with our identity, gets them delivered and we have to pay for it)</li> </ol> </li> <li>• Crime:               <ol style="list-style-type: none"> <li>1. Bogus firm (firm does not really exist)</li> <li>2. Fraud (e.g. order is confirmed, invoice has to be paid, but goods are never delivered)</li> </ol> </li> <li>• Uncertain legal status (if something goes wrong, can we accuse the provider?)</li> </ul>	<ul style="list-style-type: none"> <li>• Higher logistics cost (goods have to be sent to the customer's location)</li> <li>• Anonymity of customers (how to make targeted advertisements?)</li> </ul>

**Table 4: Disadvantages of E-Commerce.**

## 6. Conclusion

E-Commerce is not just about conducting business transactions via the Internet. Its impact will be far-reaching, and more prominent than we know currently. This is because the revolution in information technology is happening simultaneously with other developments, especially the globalization of the business. The new age of global e-commerce is creating entirely new economy and that will tremendously change our lives, will reshape the competition in various industries, and alter the economy globally. As companies are gaining high profits, more and more other companies are developing their websites to increase their profits. Since more businesses are being held online resulting in high economy development and emergence of a more innovative and advanced technology.

As the trend of on-line transactions continues to grow, there will be an increase in the number and types of attacks against the security of on-line payment systems. Such attacks threaten the security of the system, resulting in systems that may be compromised and less protected, resulting in consumer privacy issues. Consumers may be at the risk for losing their personal data, since they may be unaware of the security aspect of performing on-line transactions. Therefore, it is very important to make the Internet safe for buying and selling the products on-line. Global privacy consistency is required, as Internet usage is largely unregulated, which means that the laws in one country are not aligned with the laws in other countries.

## **Chapter 02: E-Payment Security**

## 1. Introduction

The Internet's openness makes it the perfect platform for e-commerce, it offers an inexpensive mass communication media and an economy of scale for low-cost distribution. However, the lack of security of web-based transactions and the ease with which the privacy of online communications can be violated are e-commerce's main stumbling blocks. Internet's very openness means that all communication traveling over it is inherently difficult to secure. To make matters worse, hacking is an epidemic that is on the rise.

Here are some eye-opening figures to contemplate: A study by Gartner Inc. (is a global research and advisory firm) indicates that 50 percent of all small to midsize enterprises were hacked in 2003, with almost 60 percent of those not even knowing they had been hacked.

In this chapter, we will show the security services or requirements, the different security models for payment using card, the most famous protocols used in online shopping, a detailed study of 3D Secure System and a summary of two alternatives that implement 3D Secure scheme.

## 2. Security Services

A security service is a processing or communication service that is provided by a system to give a specific kind of protection to system resources. They may be stated as follows [5]:

### 2.1. Access Control (AC)

We have chosen two definitions to this service:

- AC is a Protection of system resources against unauthorized access.
- AC is a process by which use of system resources is regulated according to a security policy and is permitted only by authorized entities (users, programs, processes, or other systems) according to that policy.

This security service protects against a system entity using a system resource in a way not authorized by the system's security policy.

### 2.2. Authentication

The process of verifying a claim that a system entity or system resource has a certain attribute value. This security service verifies an identity claimed by or for an entity.

### 2.3. Authorization

A process for granting approval to a system entity to access a system resource. Some synonyms are "permission" should be used in *role-based access control* context and "privilege" should be used in *computer operating systems* context, and in the *PKI* context we should use "authorization".

### 2.4. Availability

The property of a system or a system resource being accessible, or usable or operational upon demand, by an authorized system entity, according to performance specifications for the system; i.e., a system is available if it provides services according to the system design whenever users request them. This security service protects a system to ensure its availability.

## 2.5. Data Confidentiality

We have chosen two definitions to this service:

- The property that data is not disclosed to system entities unless they have been authorized to know the data.
- The property that information is not made available or disclosed to unauthorized individuals, entities, or processes

This security service protects data against unauthorized disclosure.

## 2.6. Data Integrity

The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. This security service protects against unauthorized changes to data, including both intentional change or destruction and accidental change or loss, by ensuring that changes to data are detectable.

# 3. Card payment security models

Based on the payment system of bank, there are many models [6], we present in this document four models:

## 3.1. No security model

Its features: users complete control of the bank card business information, the transmission of messages without bank card security. See figure 1.

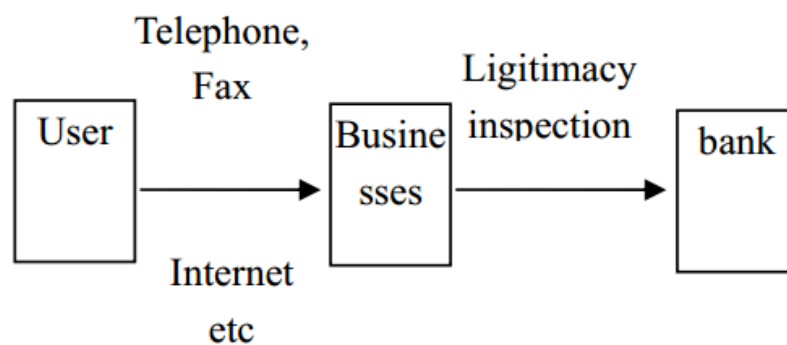


Figure 1: No security model. [6]

## 3.2. Through third-party brokers paid model

Its characteristic is as follows: Bank card information is not open to the transmission network, is paid by users. Both businessmen trusted third party (agents) to complete. See figure 2.

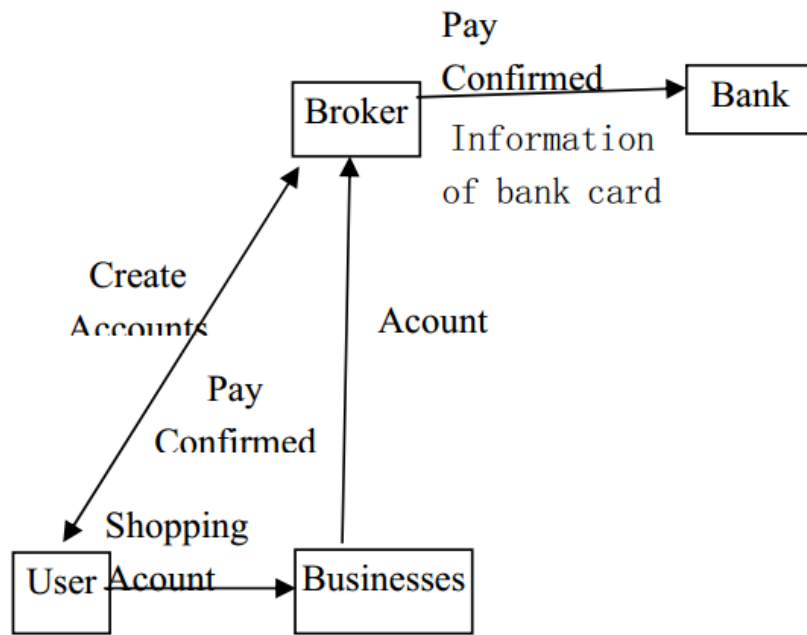


Figure 2: Through third-party brokers paid model. [6]

### 3.3. Simple encrypted payment system model

Its characteristic is as follows: the use of encryption technology to bank cards and other critical information encrypted digital signature to confirm the authenticity of the message. Business servers and the need for software support services. See figure 3.

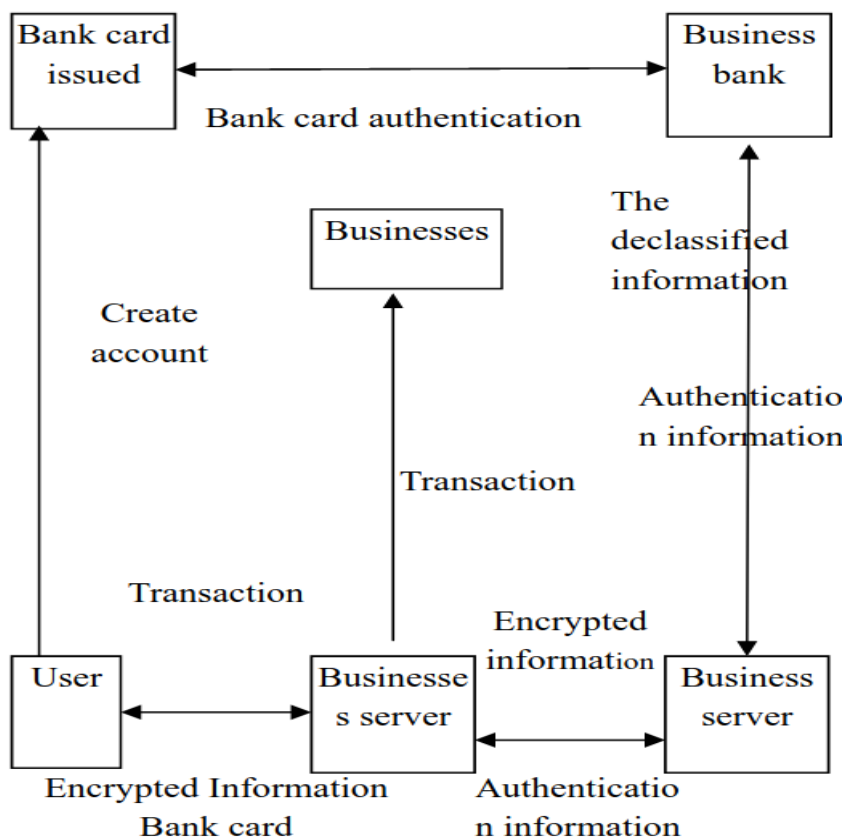


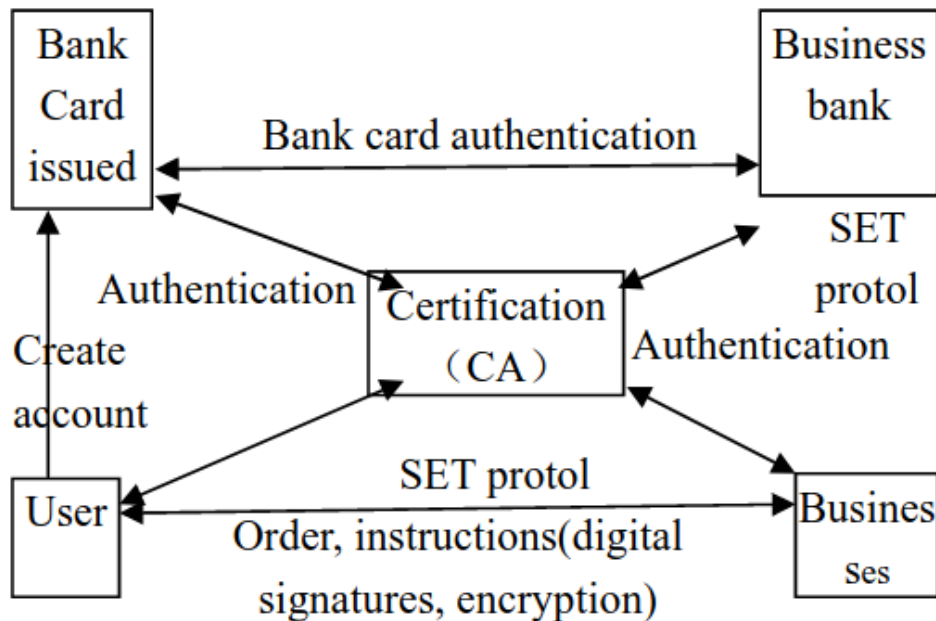
Figure 3: Simple encrypted payment system model. [6]



### 3.4. SET (security electronic transaction) model

"Secure Electronic Transactions," and referred to the SET. In an open Internet is a realization of the international agreements and standards for secure electronic transactions.

Their characteristics are as follows: SET transactions participants provide the certification to ensure data security, integrity and non-repudiation of transactions, in particular to ensure that no information leaked to the cardholder's account for the businesses guarantee and the safety of the SET.



**Figure 4: SET (security electronic transaction) Model. [6]**

Such a system more suited to the B to C mode of transaction. Consumer adoption of a way to strengthen the security of the system has also lost the anonymity feature unable to protect consumer privacy.

## 4. Security protocols

Internet is like great black hole when giving credit card information into it. So, the security needs to be implemented without any doubts. Customers must be able to select a mode of payment, on the other hand must verify their ability to pay. In this section, we have two most popular security protocols, and the typical transaction process [7].

### 4.1. Most popular protocols

There are several protocols defined for secure ecommerce transaction, and most famous are SSL and SET.

#### 4.1.1. Security Socket Layer protocol SSL

Is a cryptographic protocol, designed to provide a secure communications over a computer network, it encodes the whole session between computers and provides the safe communication service on Internet. It is widely used in many applications such as web browsing, email, instant messaging, voice over IP, and of course e-commerce transactions.

SSL Protocol was developed by Netscape Communications Corporation. There are two layers that compose SSL. At the lowest level, developed on top of TCP, there is the SSL Record Protocol which receives non interpreted data from higher layers. The SSL Record Protocol is used for the encapsulation of various higher level protocols.

The independent of an application protocol and SSL is the great advantage of this protocol. A higher level protocol can be built on top of the SSL Protocol transparently. The major disadvantage of SSL is that it cannot prevent personal information from being violated. In addition, the merchant can view or tamper this information. A comparisons between SET and SSL can be found in the next table.

#### 4.1.2. Secure Electronic Transaction (SET)

It was incorporated by Visa and MasterCard with participation from several technology companies including IBM and Microsoft. By using SET, this means that your entire credit card number is never travelling across the net- rather pieces of it- and that no human eye sees the entire card number. SET supports electronic commerce security based on Certificate Authority (CA), in Algeria, the certificate authority is AGCE (Autorité Gouvernementale de Certification Electronique).

SET is a common secure payment protocol used in e-commerce. Five parties, namely, customer, seller, payment gateway, certificate authority and issuer, are involved in the payment process. Although SET is secure to make online transactions, it is not recommended for micropayment because it is considered to be time consuming, because of the several parties involved. Besides, other disadvantage is that all parties may have to authenticate themselves.

	SSL	SET
<b>Protocol Type</b>	Secure communication protocol (end to end)	Secure payment protocol (multi party)
<b>Entities</b>	Buyer to seller	C, M, PG
<b>Authentication</b>	Only merchant authentication	Mutual authentication
<b>Privacy</b>	No privacy from merchant	Good: by using dual signature
<b>Ease of Use</b>	good: convenience	Consumer credit card certification required
<b>Mobility</b>	Good: can be used on any machine	Fair: restricted on computer installed SET certification
<b>Efficiency</b>	Good:	Fair: due to the complex cryptography
<b>Popularity</b>	Very adopted	Not very adopted

**Table 5: Comparison of SSL and SET protocols. [7]**

## 4.2. Typical transaction process

There is not only one way to do the e-commerce transaction, but typical there eCommerce has following elements:

- **Advertising:** the company communicates its products and services i.e. makes a catalogue
- **Offering:** the company offers specific goods and services,

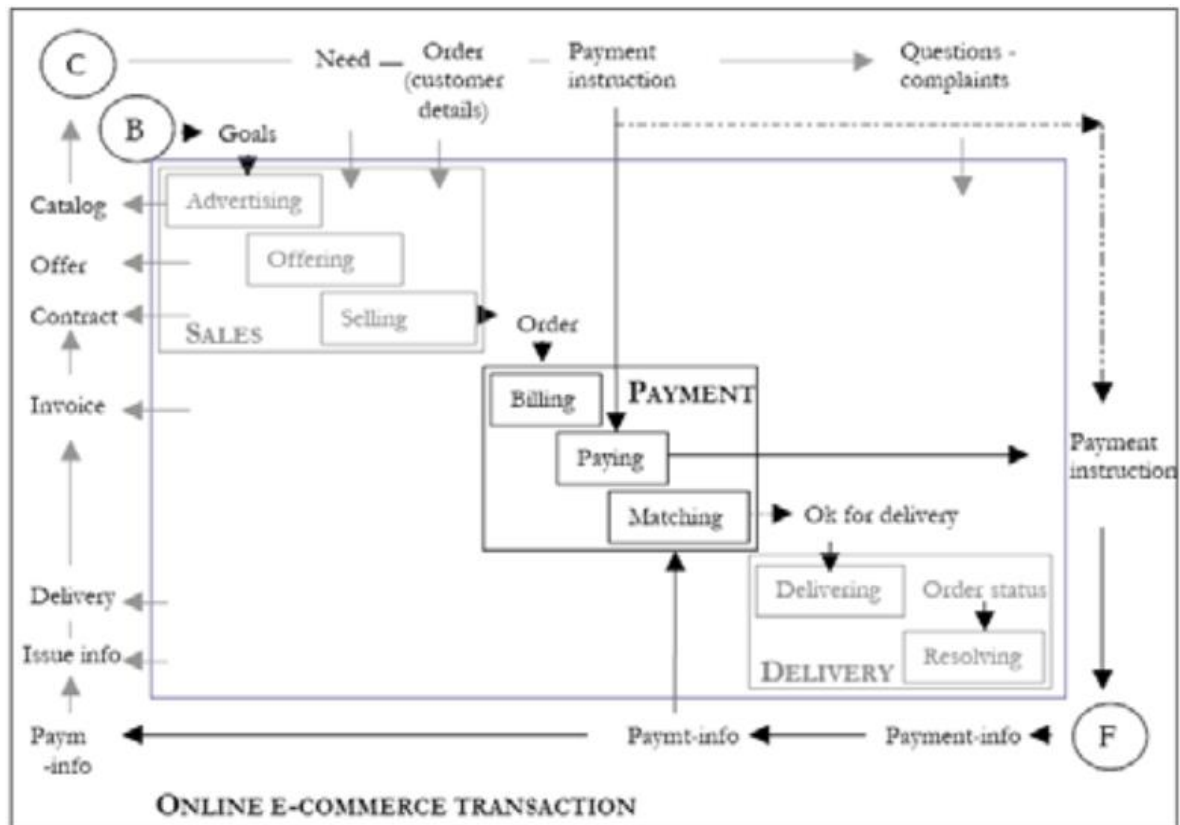


Figure 5: Typical transaction process. [7]

- **Selling:** the company agrees with the customer on the content of a specific order
- **Billing:** the company produces the invoice,
- **Paying:** the buyer pays the seller by giving a payment instruction,
- **Matching:** where the seller matches the payment information like the authorization results and the actual crediting of account.
- **Delivery:** where the seller delivers merchandise to the buyer.
- **Resolving:** the seller and buyer try to resolve delivery or payment issues related to the purchase.

This could be considered as one of the basic transaction flow chart, but it is up to seller to decide how he wants use e-commerce process (and of course buyer to accept it).

## 5. The 3D Secure System

In 2001 [8], and after the failure of SET, Visa and MasterCard began the development of two independent schemes designed to improve the security of payment card-based e-commerce.

The primary goal of both schemes was the authentication of the cardholder in order to reduce Internet-based CNP<sup>1</sup> fraud. Visa introduced 3D Secure -branded by Visa as the 'Verified by Visa' scheme- while MasterCard introduced the Secure Payment Application (SPA). Despite initial objections to 3D Secure, MasterCard eventually abandoned the full-scale implementation of SPA, and adopted 3D Secure under the brand name of 'MasterCard SecureCode'.

As stated above, the primary goal of 3D Secure is to authenticate the cardholder during a payment transaction in order to reduce CNP payment card fraud. The authentication of a cardholder is what's missing in a typical Internet-based CNP transaction (unlike the CP equivalent, where the cardholder is present with the card and can be required to perform one or more cardholder verification methods, such as signing a receipt, or entering a PIN).

3D Secure requires that cardholders 'enrol' in an issuer-managed service, either while making a purchase online, or in advance. The cardholder will typically be asked to choose a password as well as a personal assurance message. During a purchase transaction, the cardholder will be prompted to enter their 3D Secure enrolment password in order to 'prove' that it is in fact the legitimate cardholder making the transaction, and not another party fraudulently using the cardholder's details. The enrolment credentials are kept completely separate from the payment card and merchant systems and so should not be vulnerable to casual observation or collection (as is the case with the CVV2 value).

Visa refers to this process as 'Payment Authentication', although this is technically a misnomer since the payment itself is not authorized or authenticated during cardholder authentication in 3D Secure. Once cardholder authentication is complete, payment authorization occurs via the normal merchant acquirer path using a payment card brand proprietary network (e.g. VisaNet or Banknet) to submit an authorization request to the acquirer for settlement. MasterCard correctly refers to the 3D Secure component of a payment transaction as 'Cardholder Authentication'.

### 5.1. Architecture:

The '3-D' in 3D Secure refers to the 'Three Domain' model of the scheme, which includes:

- **Issuer Domain:**

This domain includes the cardholder and their card-issuing bank. In the issuer domain, the issuer manages the enrollment of the cardholder into the scheme as well as the authentication of the cardholder during a purchase.

---

<sup>1</sup> Visa defines a 'card not present' (CNP) transaction as: "... a transaction that takes place remotely - over the internet, by telephone or by post."

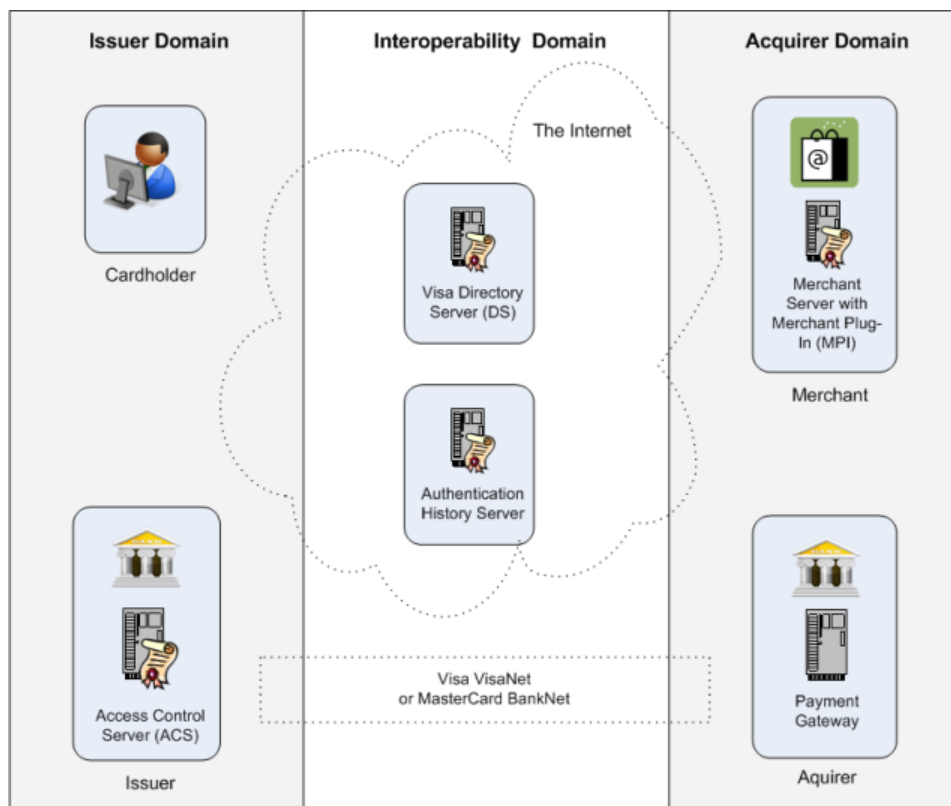
- **The Acquirer Domain:**

This domain includes the merchant and their acquiring bank. The acquirer provides transaction processing services and ensures that the merchants are operating under the agreement of the scheme.

- **The Interoperability Domain:**

This is a conceptual domain that describes the ‘interconnect’ between the issuer and acquirer domains. As we’ll see below, a unique feature of the Interoperability Domain is that it relies on the Internet in addition to the traditional and proprietary payment card networks.

The next figure illustrates an overview of the scheme’s architecture and components.



**Figure 6: 3D Secure Architectural Overview. [8]**

There are three core requirements for the successful initiation of a 3D Secure authentication attempt. These are:

1. The first is that the card issuer must implement an Access Control Server (ACS), including choosing an enrollment and authentication strategy. The payment card brand (Visa or MasterCard) may establish region-specific rules that require issuers to use specific authentication strategies.

2. The second is that the merchant (or services acquired by the merchant) must implement a merchant plug-in (MPI), allowing the merchant to determine if the cardholder is enrolled in 3D Secure, and if so, initiate the 3D Secure cardholder authentication process.

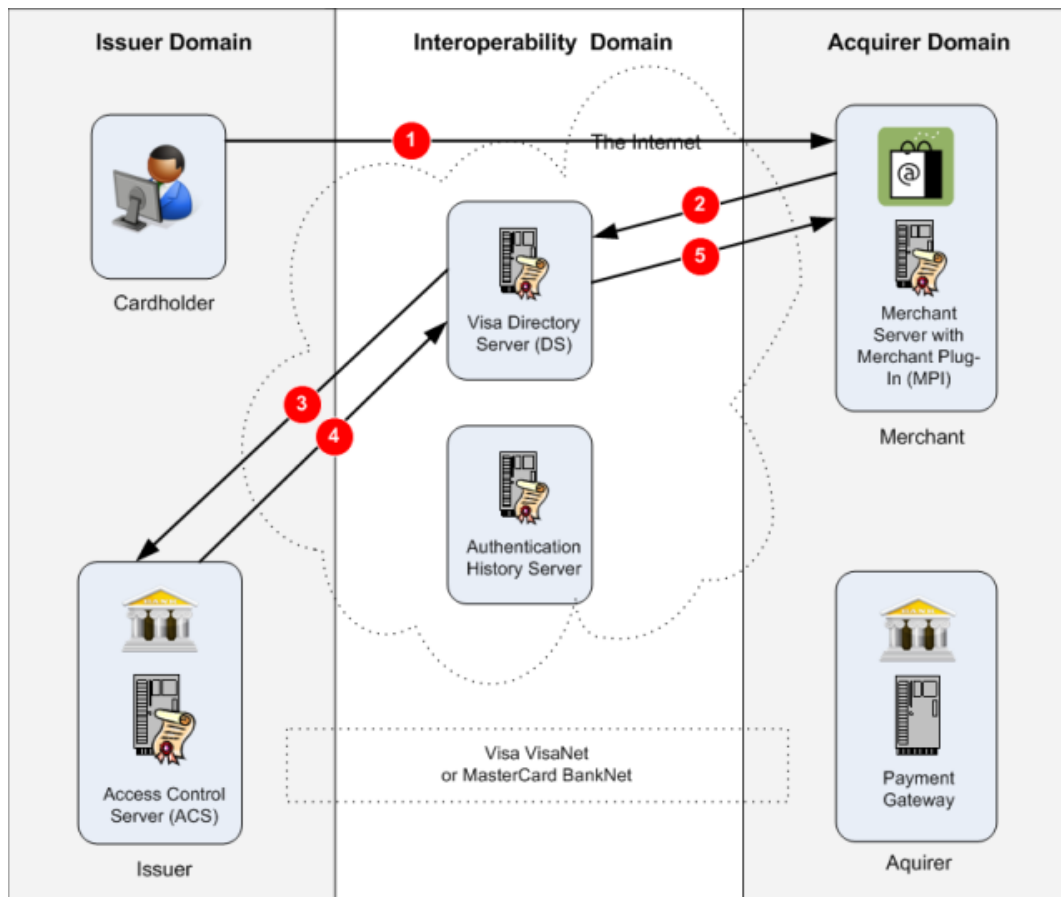
3. The third is that the cardholder must be enrolled in 3D Secure. Users may be asked to enroll ‘on the spot’ -in what is referred to as ‘activation during shopping’ or Activation Anytime as part of the payment transaction- or they may be asked to enroll in advance at the issuer’s site. Authentication schemes include static passwords, chip and PIN (via a portable reader), and even one-time passwords (OTP) sent via SMS to the cardholder’s mobile phone.

## 5.2. Typical transaction process of 3D Secure

There are two phases in the 3D Secure authentication process. The ‘Verify Enrolment’ phase and the ‘Cardholder Authentication’ phase.

### Phase 1: Verify Enrolment

As illustrated in Figure 7 below, during the verify enrolment phase, the merchant will attempt to determine if the cardholder is enrolled in 3D Secure. Steps 1-5 above are performed as follows:



**Figure 7: Verify Enrolment. [8]**

1. The customer browses the merchant’s site, selecting items to purchase and then attempts to complete the purchase by beginning the ‘check-out’ or payment process. The customer selects a payment card as their payment method and enters their payment card details.

2. Having received payment card details, a 3D Secure-enabled merchant will attempt to verify the enrolment of the payment card in 3D Secure. 3D Secure-enabled merchants will implement a merchant plug-in (MPI). The MPI may be implemented directly by the merchant,

or by a payment gateway or service provider. The merchant (or service provider), using the MPI, will attempt to contact the Visa Directory Server

(DS) located in the Interoperability Domain via the Internet. The MPI will send a Verifying Enrolment Request (VEReq) to the DS which includes the primary account number (PAN) of the cardholder. The MPI will be required to authenticate with the DS using certificates or a merchant ID and password. The MPI will communicate securely with the DS using SSL/TLS.

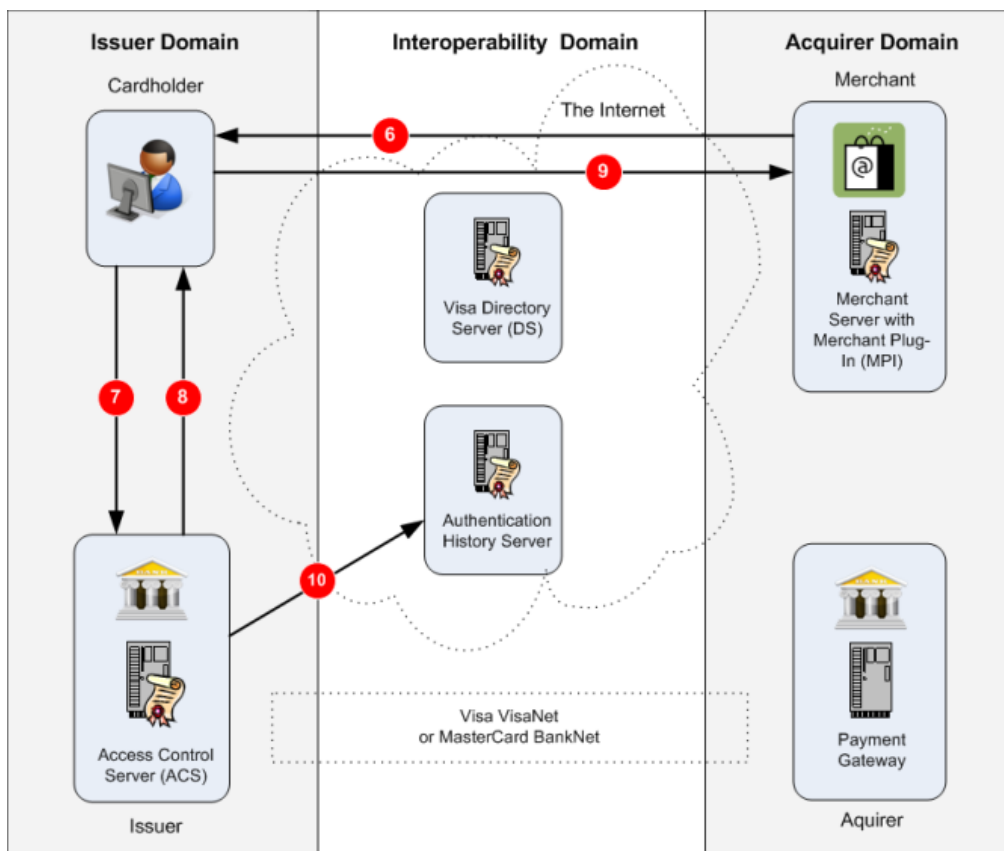
3. Based on the PAN, the DS will contact that card issuer's Access Control Server (ACS) in order to determine whether the PAN is enrolled in 3D Secure. The DS will authenticate itself to the ACS using the scheme brand root certificate and SSL/TLS.

4. The ACS will respond to the DS, indicating whether the PAN is enrolled in the scheme.

5. The DS will respond to the MPI with a Verifying Enrolment Response (VERes) message, indicating to the MPI whether the PAN is enrolled in the scheme or not. The VERes message will also include the URL of the ACS if the cardholder is enrolled.

## Phase 2: Cardholder Authentication

As illustrated in Figure 8 below, if the cardholder's PAN (primary account number) is enrolled in 3D Secure, the merchant will attempt to initiate cardholder authentication.

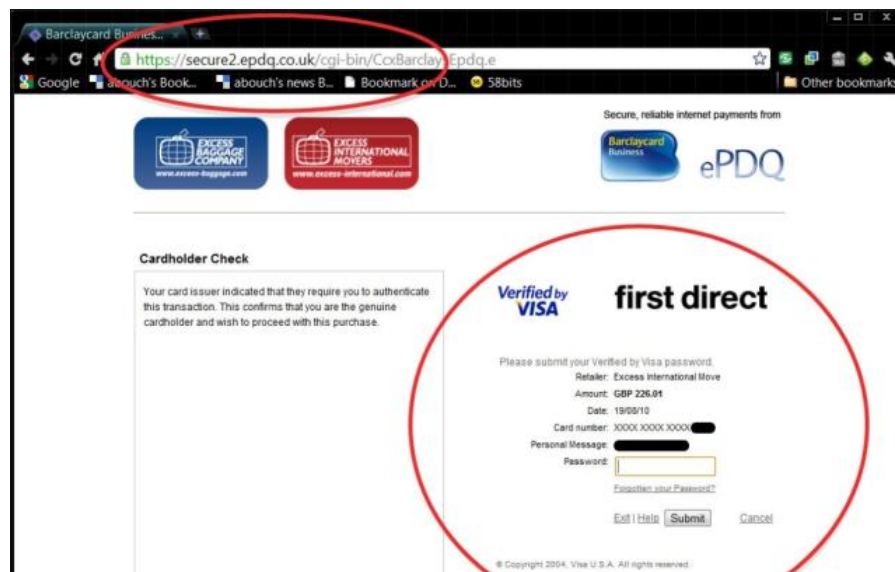


**Figure 8: Cardholder Authentication. [8]**

6. Using the MPI, the merchant will create a Payer Authentication Request (PAREq). This is a signed request, including the PAN and ACS URL. The merchant will send a specially



formatted web page to the customer's browser. This page will typically contain an iFrame, which is a web page within a web page that is capable of loading content from a URL that is independent of the main URL shown in the browser address bar.



**Figure 9: Merchant Web Page With 3D Secure IFRAME. [8]**

Figure 9 above shows a merchant web page with an embedded iFrame, showing the 3D Secure ACS (Verified by Visa) authentication request. In this case, the merchant is using a payment gateway and a hosted service. The iFrame has been given the URL to the ACS server and is populated with the content from the ACS server of the card issuer.

Early implementations of 3D Secure used pop-up windows to show the 3D Secure ACS authentication page, however pop-ups have now been specifically forbidden by both MasterCard and Visa.

7. The customer enters their Verified by Visa or SecureCode password and submits the form contained within the iFrame to the ACS. Both the user's credentials and the MPI PAREq are submitted to the ACS.

The personal assurance message, which was chosen during enrolment, is also shown on the ACS authentication page and is designed to reassure the user that the page they are looking at is in fact an authentic 3D Secure ACS request.

8. In response to the submitted form above, the ACS prepares a Payer Authentication Response (PAREs) message which is sent back to the customer's browser.

9. The PAREs is then forwarded to the MPI via the customer's browser. The MPI verifies the signature and response of the PAREs. The transaction status of PAREs is used to determine whether the customer has successfully authenticated with 3D Secure. A combination of the transaction status in PAREs and scheme rules will determine whether the merchant can proceed with a payment authorization request. If the merchant proceeds with a payment authorization request, the transaction status from PAREs will be carried forward into scheme-specific fields and included in the merchant payment authorization request. The transaction status results from PAREs are:



- a) “Y”: password correct
- b) “N”: password incorrect
- c) “U”: it was not possible to validate the password, for example because of a 3D Secure system component failure.
- d) “A”: proof that the merchant attempted to initiate an authentication attempt.

10. The ACS sends a record of the authentication attempt to the Authentication History Server.

Figure 10 shows a 3D Secure authentication page requesting an OTP that will be sent to a user’s mobile phone.

While the objectives of 3D Secure are clear, its impact and effectiveness in preventing CNP, based fraud might be less so.

The following is a review of 3D Secure including advantages and disadvantages from the merchant, acquirer, issuer and cardholder perspectives.

Verified by  
**VISA**

**Protecting your online payments**

One-Time Passcode is required for this purchase. This passcode has been sent to your registered mobile \*\*\*\*\*9469

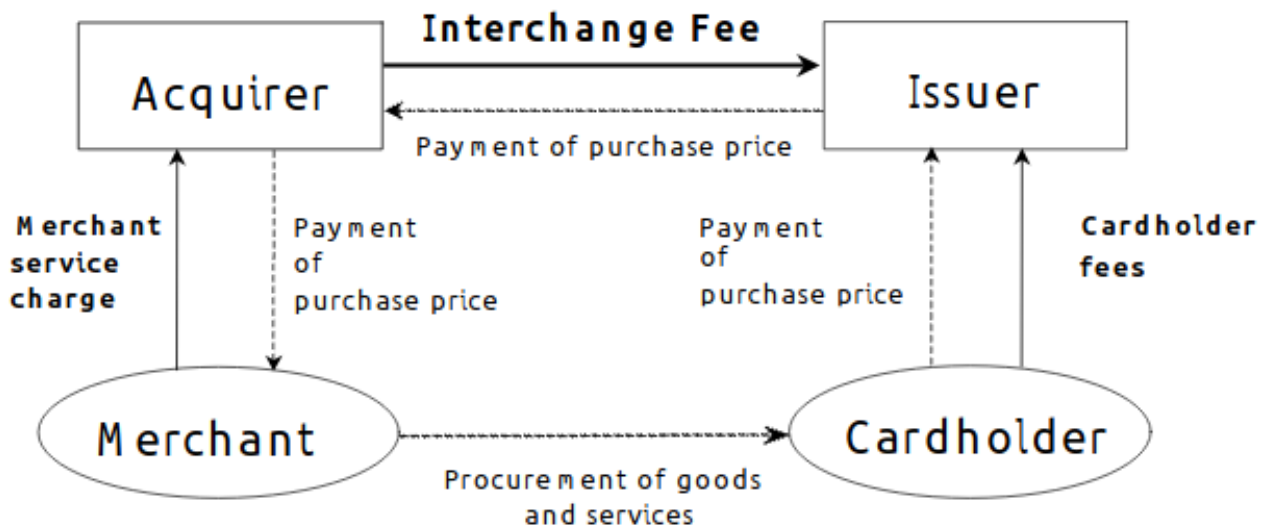
Merchant GOLFSTORE 3DS  
Amount USD 45.99  
Date 17:10:09  
Card Number XXXX XXXX XXXX 0622  
Reference Id 299879

Enter One-Time Passcode  ⓘ

I agree that by clicking the box I have read, understood and accepted the 3D Secure Terms and Conditions.

**Figure 10: 3D Secure Authentication Using A One Time Password Via SMS. [8]**

### 5.3. Four-party of 3D Secure system



**Figure 11: Four party system. [8]**

#### 5.3.1. The Merchant

It's arguable that the entity most significantly affected by 3D Secure is the merchant.

The overriding objective of the merchant is to successfully sell the products or services they are advertising on their website. Any process, procedure or security measure that the merchant implements must be considered within the context of this objective.

The question then is what effect does 3D Secure have on the merchant's business?

The objective of 3D Secure is to reduce CNP fraud. So, in theory, a reduction in fraud should also mean a reduction in chargebacks to the merchant and therefore an increase in revenue (via a reduction in losses).

However, the merchant must consider several factors when deciding whether to implement 3D Secure, including the overall cost of implementation as well as the potential for 3D Secure to negatively impact sales. Areas that are outside the merchant's control also deserve special attention.

The single greatest advantage to the merchant in implementing 3D Secure is the policy-based reduction in chargebacks. This means that if the merchant has implemented 3D Secure, and cardholder authentication was attempted via 3D Secure, the merchant will be guaranteed the payment. The payment will not be eligible for dispute or chargeback by the cardholder via the issuer. The liability for a fraudulent transaction -where 3D Secure authentication was either attempted or succeeded -shifts from the merchant to the issuer.

The potential disadvantages to the merchant from implementing 3D Secure are significant and include the risk of shopping cart abandonment. The cost of implementing the MPI, including application level changes to the merchant's website

### **5.3.2. The Acquirer**

Acquirers do not directly participate in 3D Secure. An acquirer will receive an authorization request 'after' an attempted 3D Secure authentication is complete. The acquirer will process the payment authorization request as per the usual process via a proprietary payment card network such as VisaNet or Banknet.

Acquirers do benefit from 3D Secure in terms of reduced interchange fees, as well as reduced administrative costs in handling disputed transactions and chargebacks where transactions have qualified as guaranteed payments, although these costs are typically passed on to the merchant.

### **5.3.3. The Issuer**

3D Secure impacts the issuer in two distinct ways:

1. The issuer must implement the ACS. The issuer is therefore also responsible for cardholder communication, awareness and user experience as well the implementation of appropriate security controls.

2. The liability for qualifying and fraudulent payment card transactions that have been authenticated via 3D Secure shifts away from the merchant and onto the issuer.

The most significant advantage to the issuer in the use of 3D Secure is the protection of the 'credit card brand'. Another significant advantage is the reduction in administrative costs for disputed transactions with the acquirer.

There are significant disadvantages to the issuer in implementing 3D Secure which include: The costs of implementing/supporting the ACS, The potential for financial losses from unmitigated security vulnerabilities.

### **5.3.4. The Cardholder**

The overriding objective of the cardholder during an Internet-based CNP payment transaction is to successfully place an order for their selected products or services.

There are no direct economic benefits, and the cardholder is being asked to perform extra steps in the payment process. The disadvantages to the cardholder include the following: The cardholder is being asked to perform extra steps during payment processing, the cardholder may be required to authenticate 'twice', once with the merchant application, and once with 3D Secure, The cardholder must authenticate for every transaction.

## 6. Alternatives

In this section [8] we examine two alternatives to traditional payment card-based e-commerce.

### 06.1. PayPal

PayPal was created in 1999 by Max Levchin and Peter Thiel under a company named Confinity.

PayPal acts as an intermediary payment transfer service between two parties. In the case of the merchant, PayPal removes the need to implement traditional card payment gateway or payment processing services. The service that PayPal provides is a three-party system using an indirect push payment model to transfer payments from a buyer to a seller.

### 6.2. iDEAL

iDEAL is an e-commerce payment system developed by the Dutch banking community.

It has ‘PayPal-like’ features, but has been implemented as a four-party indirect push payment model. iDEAL leverages Internet banking facilities to provide customers and merchants with a scheme that allows them to transfer funds directly between banks. Traditional payment cards are not required in iDEAL.

iDEAL is owned and operated by Currence, a not-for-profit payment product company in the Netherlands whose purpose is to oversee national payment schemes. Currence was founded in 2005 by eight banks from within the Dutch banking community. iDEAL is funded via joining fees as well as annual product fees from member banks. Merchants are not required to pay any scheme related fees, although customer and merchant banks will agree interchange fees for the transfer of funds between banks that will form part of the cost of each transaction.

## 7. Conclusion

From a security perspective, 3D Secure represents the payment card industry’s largest ‘implemented’ effort to-date to tackle the problem of CNP fraud in e-commerce. In summary, powerful economic levers are being used to force merchants to implement a scheme that implies a much greater level of security than is really provided. Poor communication and ownership from issuers combined with an overreliance on ‘activation during shopping’ may have also contributed to merchant losses and scheme reputational damage. Preliminary data suggests that the scheme may be reducing levels of Internet-based fraud.

However, there are some disadvantages. For example, each extra security field added to an online form can seriously lower the number of completed transactions. Moreover, some customers might not know what 3D Secure is and they can close the browser window. This of course will lead to lost sales. You must also remember that there is an extra fee for the service.

# **Chapter 03: Design and analysis of 3D Secure**

## 1. Introduction:

The online payment process needs to be organized and secured using protocols. Effective use of such protocols depends on many factors. 3D Secure is a protocol developed by visa and MasterCard to improve the security of payment card-based ecommerce, it benefits cardholders and merchants by providing an additional layer of authentication.

The whole e-payment information are critical, sensitive and important. We need to protect it against loss, violation and attacks, hence the need and the importance to secure e-payment transactions and protocols. There are several security mechanisms to solve this problem including: authentication, encryption/decryption, SSL, SET...etc.

Traditional security mechanisms like authentication, encryption/decryption and SSL protect the online transactions against a few attacks and provide a few security requirements such as confidentiality, data integrity and authentication. However, there are other attacks still remain unsolved against the cardholders authentication. Therefore, visa and MasterCard have developed the 3D Secure protocol to reduce internet based CNP fraud by improving the authentication of cardholders.

In this chapter we will explain in detail our approach to build a 3D Secure architecture using client/server model, and we will add two-factor authentication to this protocol to improve and secure the online transaction against identity fraud attack. We will examine 3D Secure protocol components and messages, a general activity diagram of this protocol is outlined in this section, a detailed steps of this protocol, and finally we will show the 3D Secure pseudocode.

## 2. 3D Secure protocol components

This section describes the components, the systems, and the functions necessary to implement 3D Secure. Descriptions are summarized into the following domains:

- Acquirer Domain: 3D Secure transactions are initiated from the Acquirer Domain
- Interoperability Domain: 3D Secure transactions are switched between the Acquirer Domain and Issuer Domain
- Issuer Domain: 3D Secure transactions are authenticated in the Issuer Domain; Figure 14 depicts the interaction of the three domains and the components of each.

Because the implementation of the 3DS Requestor Environment may vary, the diagram purposefully does not imply a specific implementation of these components or how they interoperate. For example, the 3DS Client may communicate directly with the 3DS Server, or the 3DS Server and 3DS Requestor may be functionally combined. For a detailed description of each domain, please take a look to the EMVCo document [9].

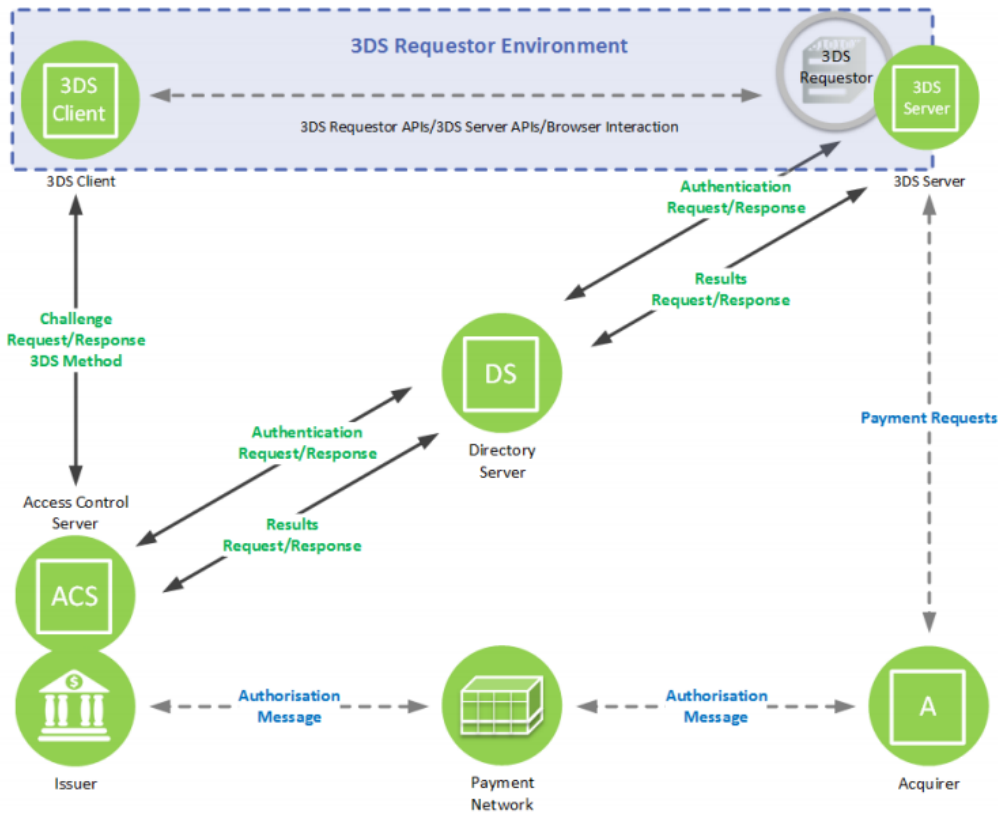


Figure 12: 3D Secure Domains and Components [9].

## 2.1. Acquirer Domain

The Acquirer Domain has the following components:

- 3DS Requestor Environment
  - 3DS Requestor
  - 3DS Client
  - 3DS Server
- 3DS Integrator
- Acquirer (for Payment Authorization)

## 2.2. Interoperability Domain

The Interoperability Domain has the following components:

- Directory Server (DS)
- Directory Server Certificate Authority (DS CA)
- Authorization System

## 2.3. Issuer Domain

The Issuer Domain has the following components:

- Cardholder
- Consumer Device
- Issuer
- Access Control Server (ACS)

## 3. Messages used by 3D Secure protocol [9]

In this section we will describe the whole messages used by 3D Secure protocol according to EMVCo standard. In our case we have developed a complete 3D secure protocol using PHP/JS with few modifications to adapt it with Algerian market. Therefore, we have not used PReq, PRes and Error messages.

### 3.1. Authentication Request Message (AReq)

The AReq message is the initial message in the 3D Secure authentication flow. The Acquirer domain (3DS Server) forms the AReq message when requesting authentication of the Cardholder. It can contain Cardholder, payment, and Device information for the transaction. There is only one AReq message per authentication.

### 3.2. Authentication Response Message (ARes)

The ARes message is the Issuer's (ACS) response to the AReq message. It can indicate that the Cardholder has been authenticated, or that further Cardholder interaction is required to complete the authentication. There is only one ARes message per transaction.

### 3.3. Challenge Request Message (CReq)

The CReq message initiates Cardholder interaction in a Challenge Flow and can be used to carry authentication data from the Cardholder.

- **Browser-based:** The CReq message is sent by the Acquirer domain (3DS Server). There is only one CReq message per challenge.

### 3.4. Challenge Response Message (CRes)

The CRes message is the Issuer (ACS) response to the CReq message. It can indicate the result of the Cardholder authentication or, in the case of an App-based model, also signal that further Cardholder interaction is required to complete the authentication.

**Browser-based:** The CRes message contains the authentication result and completes the Cardholder challenge. There is only one CRes message per challenge.

### 3.5. Results Request Message (RReq)

The RReq message communicates the results of the authentication or verification. The message is sent by the issuer domain (ACS) through the interoperability domain (DS) to the acquirer domain (3DS Server). There is only one RReq message per AReq message. The RReq message is not present in a Frictionless transaction.

### 3.6. Results Response Message (RRes)

The RRes message acknowledges receipt of the RReq message. The message is sent by the acquirer domain (3DS Server) through the interoperability domain (DS) to the issuer domain (ACS). There is only one RRes message per RReq message.

### 3.7. Preparation Request Message (PReq)

The PReq message is sent from the acquirer domain (3DS Server) to the interoperability domain (DS) to request information about the issuer's domain (ACSs) and the interoperability domain (DS). This message is not part of the 3D Secure authentication message flow.



### 3.8. Preparation Response Message (PRes)

The PRes message is the interoperability domain (DS) response to the PReq message. The acquirer domain (3DS Server) can utilize the PRes message to cache information about the issuer's domain (ACSs) and the interoperability domain (DS) (for example, about which Protocol Version(s) are supported). This message is not part of the 3D Secure authentication message flow.

### 3.9. Error Message

Error messages provide additional information about an error that occurred during message processing between the acquirer domain (3DS Server), the DS, the issuer domain(ACS), and the 3DS SDK.

## 4. Global Activity diagram

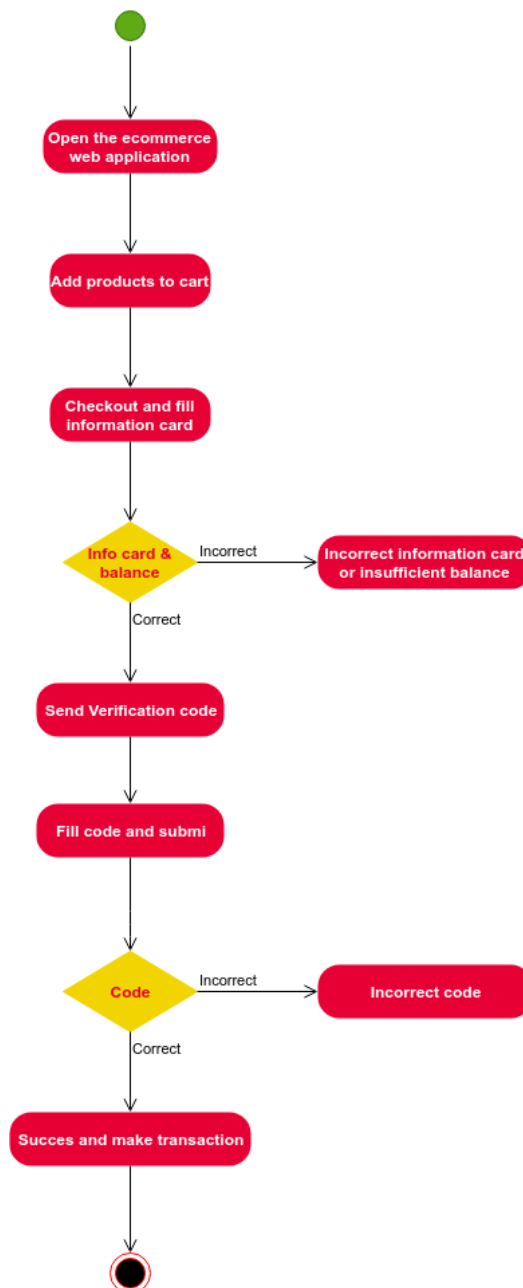


Figure 13: Global Activity diagram.

## 5. The 3D Secure process under online shopping

In this part, we will describe the successive steps of the online shopping process used by the 3D Secure protocol. In order to clarify the situation, the arrows 6-9 11 15-21 shown in the following diagram are necessary in 3D Secure protocol work, the other arrows are optional and may be varying according to what we need. In the next figure, portrays a possible flow for components within the 3DS Requestor Environment and does not preclude a specific implementation.

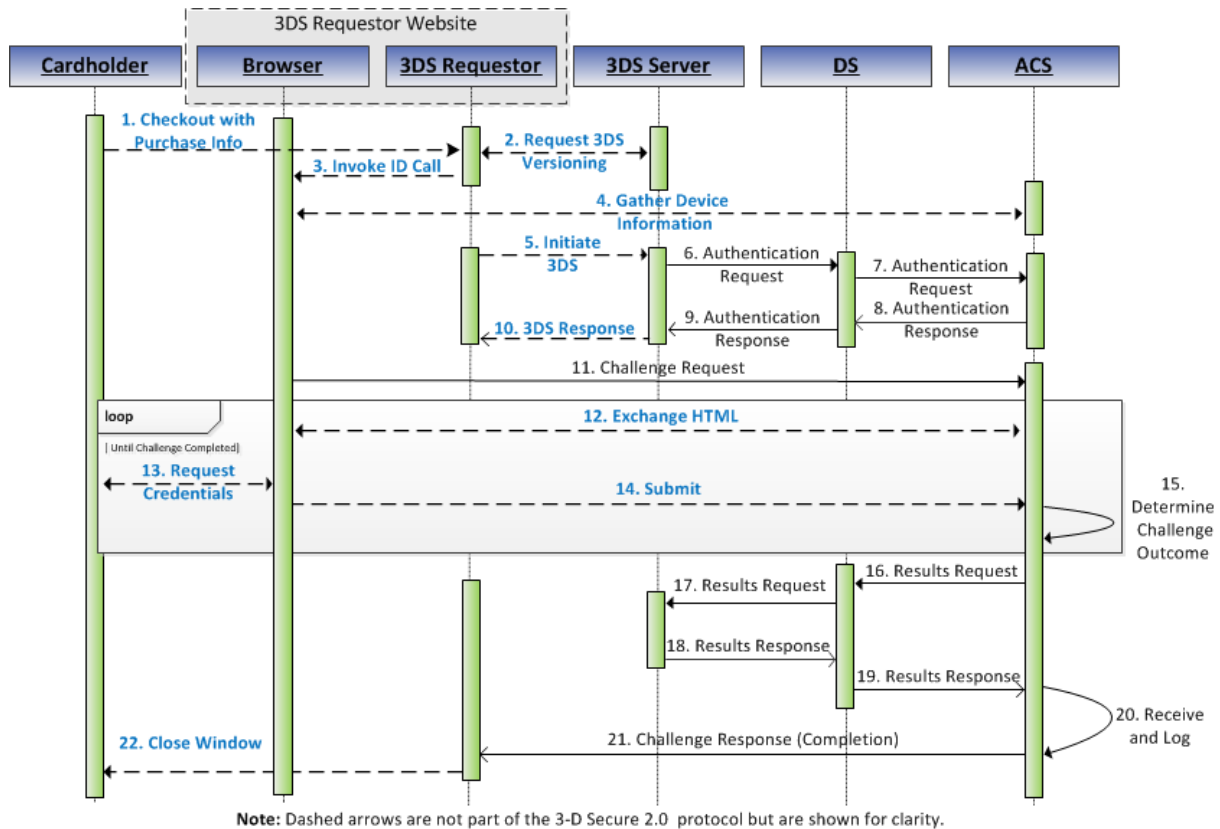


Figure 14: Sequence diagram of SD Secure process. [9]

### Step 01: Make online shopping by cardholder

To make online shopping, the cardholder (user) must login to the online shop (hanouti.dz) or create a new account, he will search to find products that help her to make the online shopping. After this he will select products and add them to the cart. Then click on the checkout button to pass to the checkout page and fill shipping information (optional), and finally choose the payment method. In our online shop, we have added a new payment method called “pgw-pfe3ds”.

We've developed an ecommerce web application to test 3D Secure protocol, This application consists of two applications combined together, and it contains many features, including what are related to the user, such as: account, product, address, order management ...etc, and some of them are related to the administrator, such as: dashboard, customer, account, category, product, order...etc.

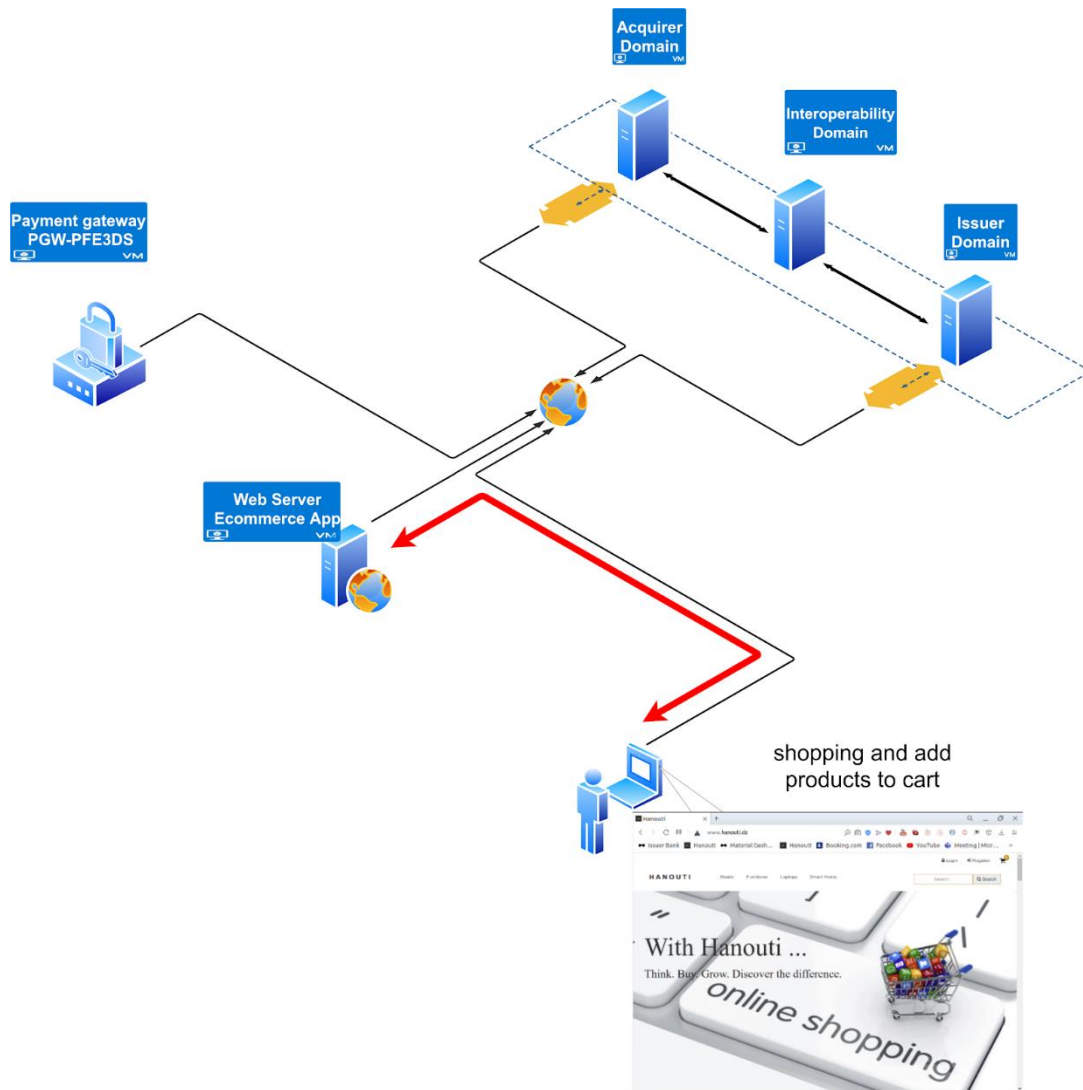


Figure 15: First step to do an online shopping.

## Step 02: 3D Secure initiation

When the user choose our payment method “pgw-pfe3ds”, the MPI (Merchant Plug-In) gather and send the merchant information including : name, website URL, total paid amount, return URL, cancel URL ...etc. to the payment gateway to initiate the 3D Secure process. After that we will automatically redirect the user to the pgw-pfe3ds web application.

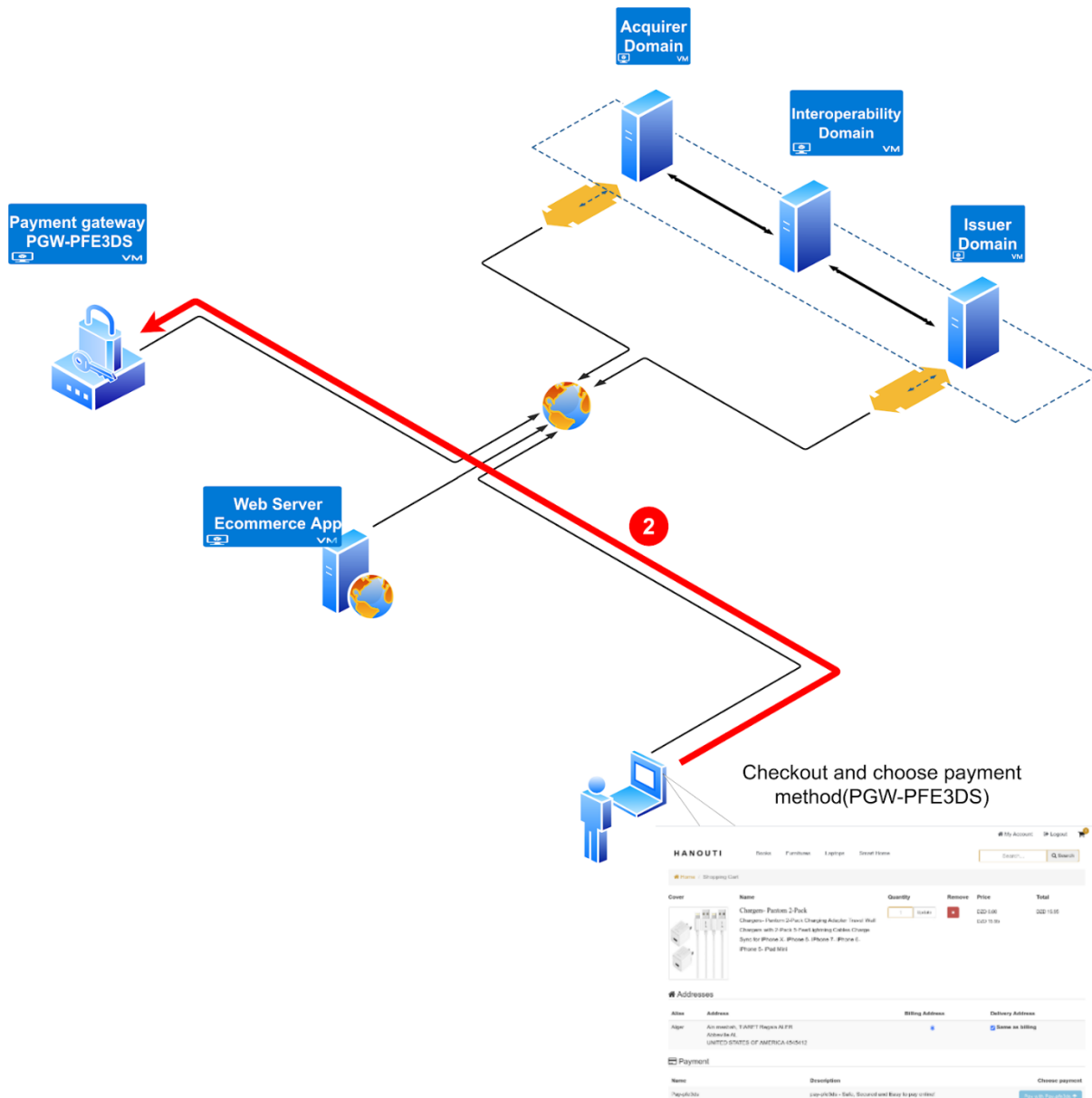
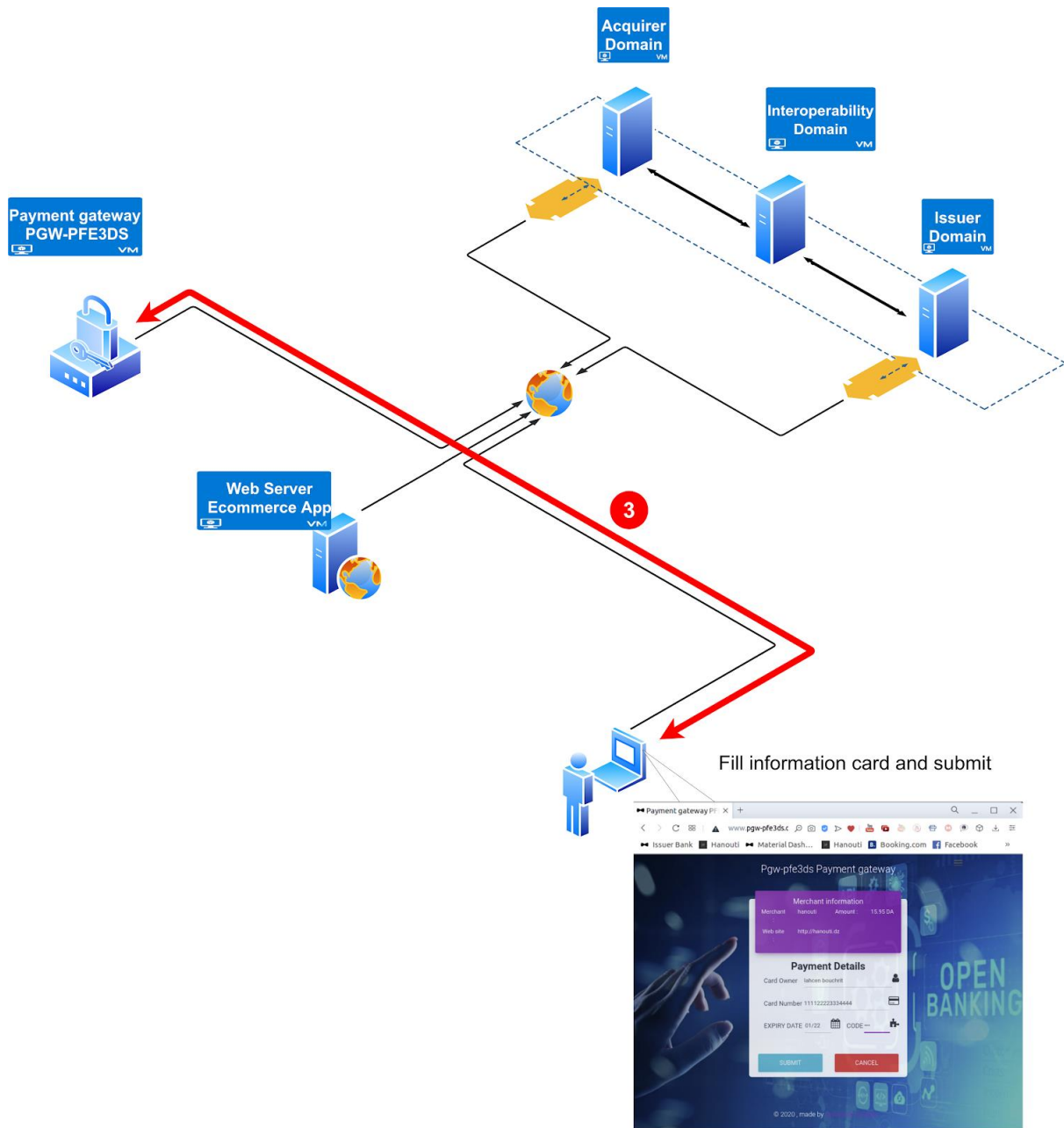


Figure 16: Second step, checkout and choose payment method.

### Step 03: Submit information card to Payment Gateway

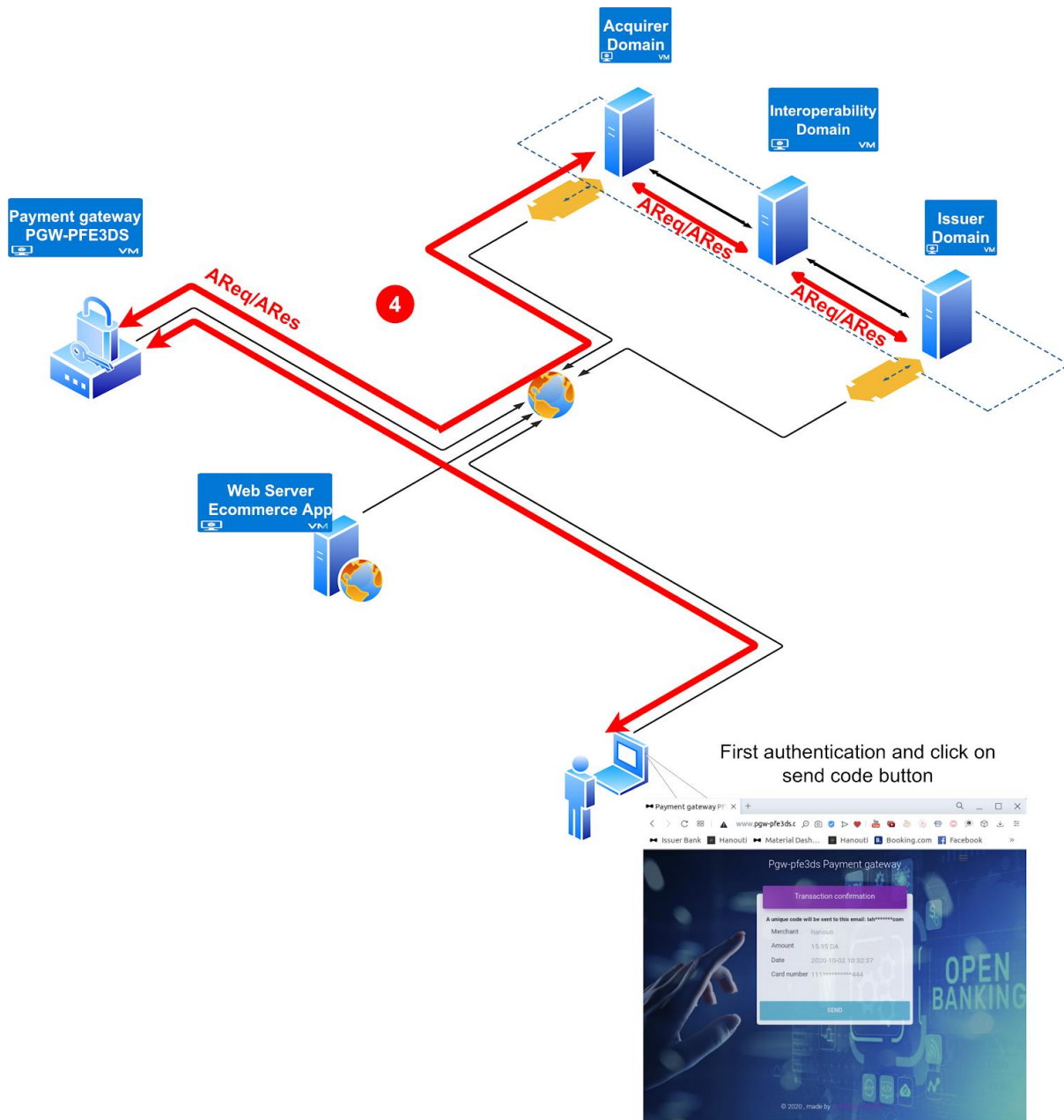
To complete the purchase and after opening the payment gateway page, the cardholder must enter and submit the correct information card. If any of this information is incomplete or incorrect, an appropriate error message will appear, also if the card balance is less than the amount of purchased products, an error message will appear.



**Figure 17: The third step, fill information card.**

#### **Step 04: Card information verification (1<sup>st</sup> Auth)**

To verify information card, the 3DS protocol uses the AReq/ARes message to authenticate the cardholder. The payment gateway sends AReq to the acquirer domain (3DS server). The acquirer domain forwards the received AReq to the issuer domain (ACS) via the interoperability domain (DS). ARes returned by the issuer domain (ACS) via the interoperability (DS) in response to an Authentication Request message (AReq) from the acquirer domain. The acquirer domain forwards the received ARes to the payment gateway. Finally, the payment gateway will proceed to the next page of the process, in this page there is a button to send OTP to the user using email, SMS or other authentication mechanisms.



**Figure 18: The fourth step, 1<sup>st</sup> authentication.**

### Step 05: Cardholder verification (2<sup>nd</sup> Auth)

After the first verification, a new page (figure 19) will appear, which is expected to contain merchant name, merchant website URL, amount, email and card number with hidden part (e.g.: abs\*\*\*\*\*ij@gmail.com, 4545\*\*\*\*\*55455), and a button to request the code. When we click on it, a CReq message will be sent directly to the issuer domain.

The issuer domain will generate and save a random number (OTP), then send it to the user using email or SMS or other mechanisms. The Cardholder enters the authentication data (OTP code) via the browser to be checked by the ACS. In response to the CReq message, the CRes message is formed by the ACS and sent to the 3DS Server to indicate the result of the authentication. If it is incorrect then an error message will appear under the code field, else the online shopping process will continue successfully.



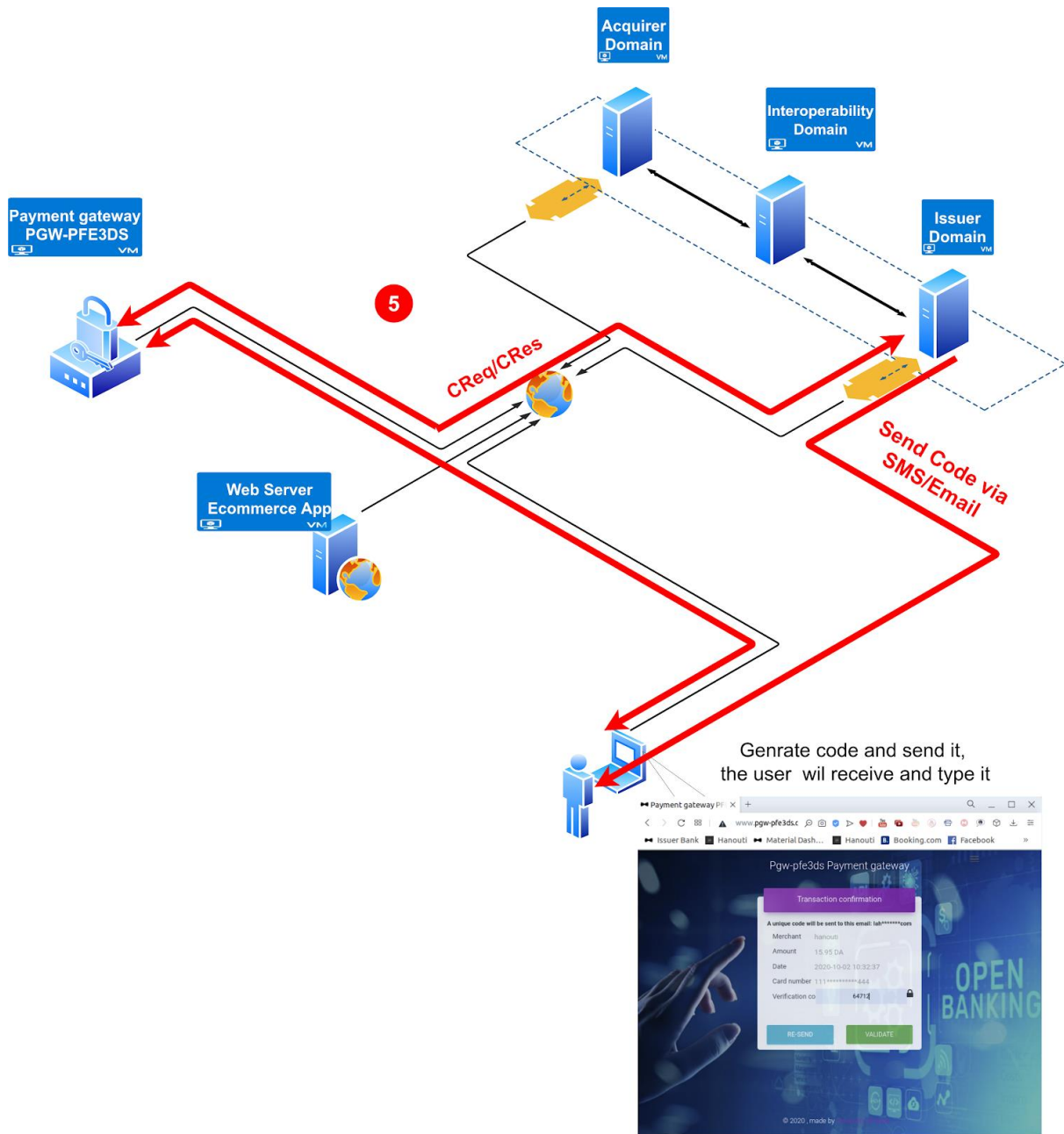
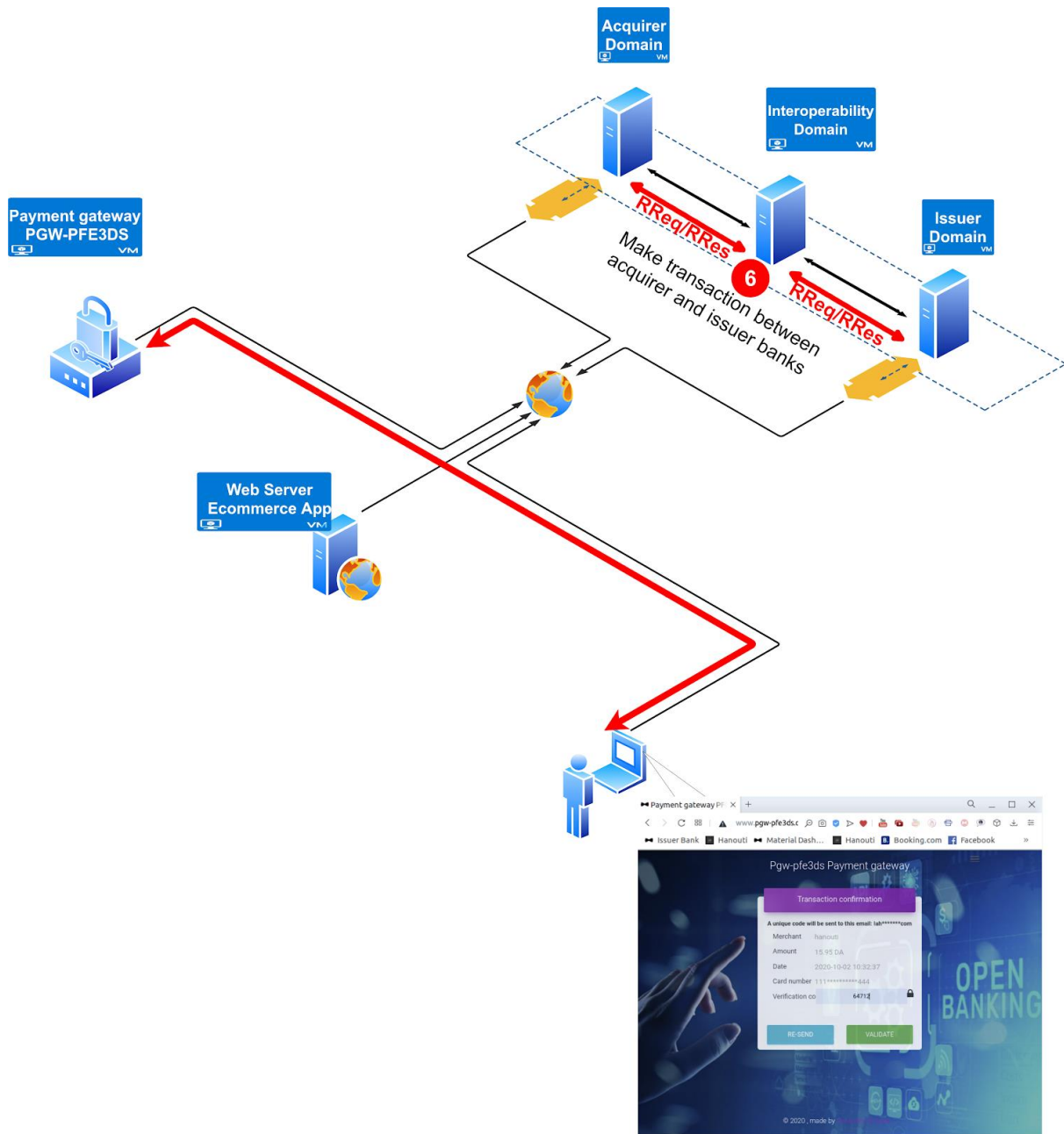


Figure 19: The fifth step, 2<sup>nd</sup> authentication.

## Step 06: Make transaction

In this step, all processes take place in the backend and the user has nothing to do with what actually happens. After the success of the second confirmation (using OTP), the issuer bank will send an RReq message (Result Request) to the acquirer bank via interoperability domain to inform him to initiate the transfer of funds from the customer's account to the merchant's account. After receiving this message by the acquirer bank, he will add the total amount paid to increase the merchant's balance and save it in the transaction log, then send a message to the issuer bank via interoperability domain to inform him about the success transaction. Issuer bank will subtract the total amount paid from the customer's account and save it in the transaction log.



**Figure 20: Bank to bank transaction.**

### Step 07: Ending process

Finally, after the success of the money transfer, the user will be redirected from the payment gateway to a page in the online shop, this page displays a copy of the online payment receipt for the transaction you have completed. The payment receipt information are the quantity and price of each product, the total amount paid, the customer information, the merchant information and the date of this shopping process. At the same time, the customer can print this payment receipt.

Proof of payment and payment information should be sent directly to the mail of the customer to ensure that they purchase these products from the merchant ecommerce website.



## 6. Pseudocode

In this part, we will use structured basic style pseudocode to describe our solution, we will show the frontend pseudocode of the end user (customer), and the backend pseudocode of payment gateway and acquirer, issuer, and interoperability domain.

### 6.1. Frontend pseudocode:

#### 6.1.1. End user (Customer):

01 Open web browser and go to online shopping webApp; 02 Authenticate. 03 Add products to cart. 04 Checkout;	}	Ecommerce Web App
05 Fill the card information. 06 Send verification code. 07 Enter the code & validate;	}	Payment gateway
08 Show payment status 09 Send payment receipt via email	}	Ecommerce Web App

### 6.2. Backend pseudocodes

#### 6.2.1. Acquirer domain

1. intrp : Interoperability;
2. iss : Issuer;
3. pgw: Payment gateway;
4. merchant: Merchant;
5. init\_message: Struct{
6.     merchant\_name : string;
7.     merchant\_webapp\_url : string;
8.     total\_paid: number
9. }
10. info\_card: Struct{
11.     cardHolder\_name: String;

```
12. card_number: Number(16);
13. exp_date: Date(MM/YY);
14. cvv: Number(3);
15. }
16. Begin{
17.   Send(pgw, init_message);
18.   wait();
19.   Receive(pgw, AReq);
20.   Send(intrp, AReq);
21.   Wait();
22.   Receive(intrp, ARes);
23.   Send(pgw, ARes);
24.   Wait();
25.   Receive(intrp, RReq);
26.   RRes = CreateRRes();
27.   Send(intrp, RRes);
28.   CreateTransaction(merchant, "Credit", AReq->amount);
29. }
```

### 6.2.2. Payment gateway

```
1. acq : Acquirer;
2. iss : Issuer;
3. code: Number;
4. init_message: Struct{
5.   merchant_name : string;
6.   merchant_webapp_url : string;
7.   total_paid: number
8. }
9. info_card: Struct{
10.  cardHolder_name: String;
11.  card_number: Number(16);
12.  exp_date: Date(MM/YY);
13.  cvv: Number(3);
14. }
15. Begin{
16.  Receive(acq, init_message);
17.  Read(info_card);
18.  AReq = CreateAReq();
19.  Send(Acq, AReq);
20.  Wait();
21.  Receive(Acq, ARes);
22.  If (IsCorrecte(ARes)){
23.    a. CReq = CreateAReq();
24.    Send(iss, CReq); // Request Code
25.    Read(code); // Entered by the user
26.    CReq->code = code;
27.    Send(iss, CReq);
```

```
27.     Wait()
28.     Receive(iss, CRes)
29.     If(CRes->validation=true){
30.         //Success shopping operation
31.         Return to Ecommerce webApp;
32.     }
33.     else{
34.         // failed shopping operation
35.         write("incorrect code");
36.     }
37. }
38. else{
39.     Write("Incorrect information card or insufficient balance");
40. }
41. }
```

### 6.2.3. Interoperability domain

```
1. acq : Acquirer;
2. iss : Issuer;
3. Begin{
4.     Receive(acq, AReq);
5.     Send(iss, AReq);
6.     wait();
7.     Receive(iss, ARes);
8.     Send(acq, ARes);
9.     Wait();
10.    Receive(iss, RReq);
11.    Send(acq, RReq);
12.    Wait();
13.    Receive(acq, RRes);
14.    Send(iss, RRes);
15. }
```

### 6.2.4. Issuer domain

```
1. intrp : Interoperability;
2. pgw: Payment gateway;
3. info_card: Struct{
4.     cardHolder_name: String;
5.     card_number: Number(16);
6.     exp_date: Date(MM/YY);
7.     cvv: Number(3);
8. }
9. Begin{
10.    Receive(intrp, AReq);
11.    ARes = CreateARes(); // information card verification
```

```
12. Send(intrp, ARes);
13. wait();
14. Receive(pgw, CReq);
15. code = GenerateCode();
16. Send(user, code);
17. wait();
18. Receive(pgw, CReq);
19. if (code == CReq->code){
20.     CRes = CreateARes({ validation = true;});
21.     Send(pgw, CRes);
22.     RReq = CreateRReq();
23.     Send(intrp, RReq);
24.     wait();
25.     Receive(intrp, RRes);
26.     CreateTransaction(user, "Debit", AReq->amount);
27. }
28. }
```

## 7. Conclusion

3D Secure is the most used protocol for the most online transactions. It is standardized by EMVCo and implemented by a large community. As we explained in this chapter, the 3D Secure protocol components are divided to three domains: issuer, acquirer, and interoperability domains. He is using seven messages to communicate between these domains, these messages are: AReq/ARes, CReq/CRes, RReq/RRes, PReq/PRes, Errors. We have also divided the online shopping process that is used 3DS protocol into seven steps. To more explaining the internal behavior of our protocol, we have written pseudocode of each domain.

There are a lot of implementations and may depend on many factors such as: banks, government, technologies used ...etc. In the next chapter we will see together our vision to implement this protocol.

## **Chapter 04: Implementation & Results**

## 1. Introduction:

We will see in the next sections what tools could be used in this project to implement the 3D Secure architecture, and what code could be integrated in the merchant ecommerce website (MPI Merchant Plug-In). Throughout this document we use the term MPI to refer to API payment solutions such as PayPal rest api ...etc.

There are many different processes involved with building a web application. In this document, we'll take a look at how the general website development process may look like. For each web application we will present the use cases, sequence diagram, class diagram, feature list, and a brief description of the source code.

To prove that our implementation of 3D Secure protocol is correct, we will show a detailed demo starting from creating accounts until coming to the end of a successful online shopping process.

## 2. Environment description

In this section, we will present requirements of this project, for both parts: software tools and hardware. We have used our laptops to satisfy the hardware requirements, and we have used open source softwares due to its advantages.

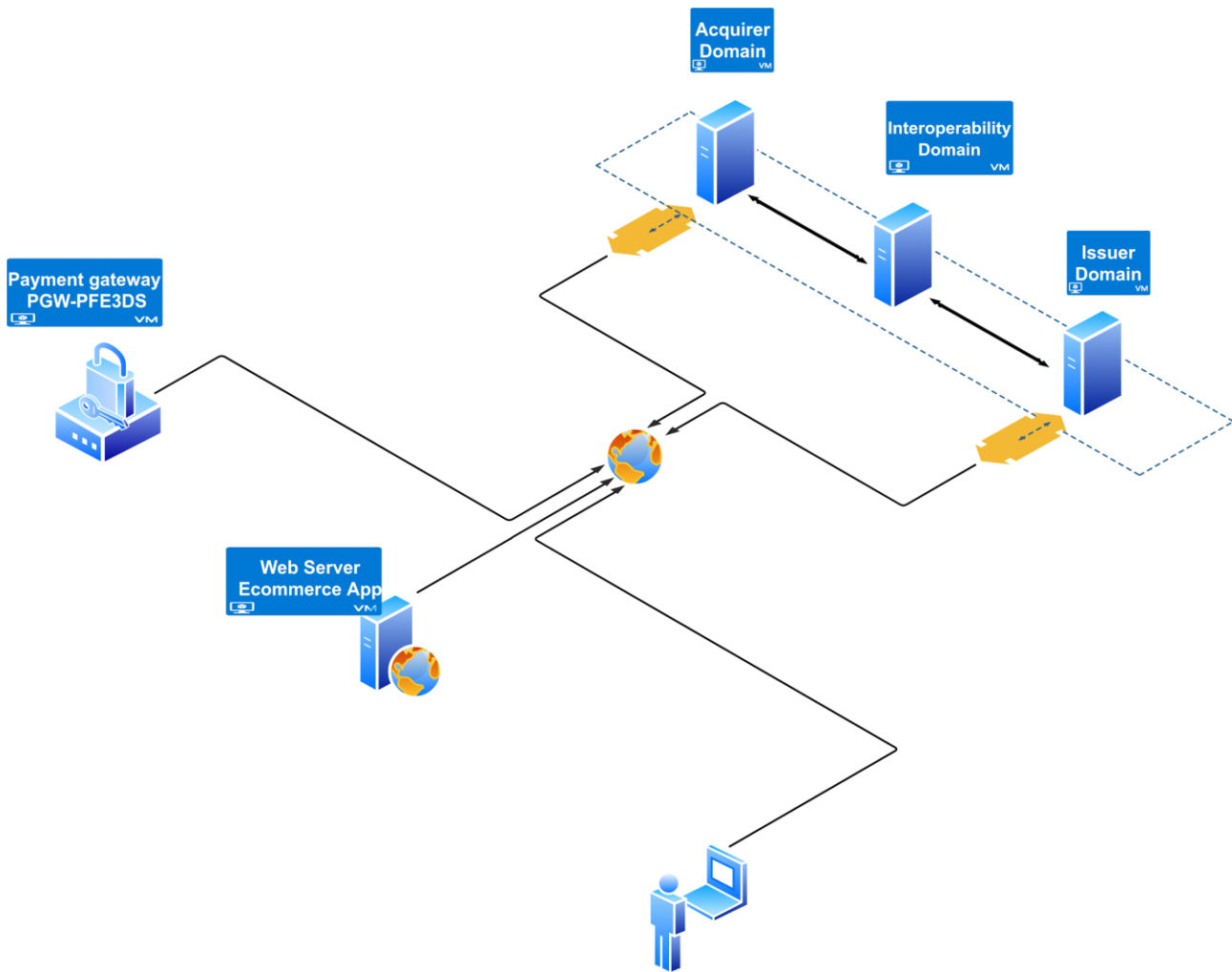
### 2.1. Hardware

The next table contains the characteristics of physical and virtual machines that we have used to implement 3D Secure protocol.

	Physical machines		Virtual machines		
Characteristic	Laptop 1	Laptop 2	Issuer	Acquirer	Interoperability
Name	Lenovo ideapad 320S	HP DV6 6B71SF	issuer	acquirer	interoperability
CPU	i5, 8#,	i7, 8#	1#(Core)	1#(Core)	1#(Core)
RAM	8 Go	8 Go	2 Go	2 Go	2 Go
DISK	SSD 256Go	HDD 1 To	256Go	256Go	256Go
Type	host	host	guest	guest	guest
OS	Windows 10	Linux 18.04 LTS Ubuntu	Linux 18.04 LTS Ubuntu	Linux 18.04 LTS Ubuntu	Linux 18.04 LTS Ubuntu

**Table 6: Characteristics of physical and virtual machines.**

This below figure shows the global architecture of our environment, there are two networks, the first one to establish a connection between different domains, where the second one to establish communication between all 3D Secure devices and components, it is the internet connection.



**Figure 21: Global architecture of 3D Secure environment.**

## 2.2. Software

The aim of this part is to present the different tools that we have working with. All our services are most often created using free tools and open source technologies.

### a) Laravel

Is an open-source web application development framework written in PHP. It is created by Taylor Otwell and released under MIT License. And it offers you rapid application development following the model-view-controller (MVC) architectural pattern. Laravel is a framework which makes it easier for you to build professional yet powerful web applications following much expressive, elegant syntax and architectural pattern.<sup>2</sup>

With Laravel the development must be an enjoyable and creative experience to be truly fulfilling. Laravel takes the pain out of development by easing common tasks used in many web projects, such as:

- Simple, fast routing engine.

<sup>2</sup> <https://www.w3adda.com/laravel-tutorial/laravel-introduction>

- Respect MVC pattern (Model-View-Controller)
- Powerful dependency injection container.
- Multiple back-ends for session and cache storage.
- Expressive, intuitive database ORM.
- Database agnostic schema migrations.
- Robust background job processing.
- Real-time event broadcasting.

Laravel is accessible, powerful, and provides tools required for large, robust applications. A superb combination of simplicity, elegance, and innovation gives you a complete toolset required to build any application with which you are tasked.<sup>3</sup>

### b) RESTful API (REST API)<sup>4</sup>

A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

An API for a website is code that allows two software programs to communicate with each other. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.

A RESTful API - also referred to as a RESTful web service or REST API - is based on representational state transfer (REST), an architectural style and approach to communications often used in web services development.

RESTful API design was defined by Dr. Roy Fielding in his 2000 doctorate dissertation. In order to be a true RESTful API, a web service must adhere to the following six REST architectural constraints:

- **Use of a uniform interface (UI).** Resources should be uniquely identifiable through a single URL, and only by using the underlying methods of the network protocol, such as DELETE, PUT and GET with HTTP, should it be possible to manipulate a resource.
- **Client-server based.** There should be a clear delineation between the client and server. UI and request-gathering concerns are the client's domain. Data access, workload management and security are the server's domain. This loose coupling of the client and server enables each to be developed and enhanced independent of the other.
- **Stateless operations.** All client-server operations should be stateless, and any state management that is required should take place on the client, not the server.

---

<sup>3</sup> <https://github.com/laravel/laravel>

<sup>4</sup> <https://searchapparchitecture.techtarget.com/>



- **RESTful resource caching.** All resources should allow caching unless explicitly indicated that caching is not possible.
- **Layered system.** REST allows for an architecture composed of multiple layers of servers.

**Code on demand.** Most of the time, a server will send back static representations of resources in the form of **XML** or **JSON**. However, when necessary, servers can send executable code to the client.

### c) Other tools

We have also used PHP<sup>5</sup> as a server scripting language, and Linux Ubuntu<sup>6</sup>, and other tools like: phpMyAdmin<sup>7</sup>, MySQL<sup>8</sup>, Bootstrap<sup>9</sup>, WampServer<sup>10</sup>, Composer<sup>11</sup>, Apache<sup>12</sup>, Visual Studio Code (vsc)<sup>13</sup>, VirtualBox<sup>14</sup>, Draw.io<sup>15</sup>, HTML<sup>16</sup>, CSS<sup>17</sup>, JQuery<sup>18</sup>, XML<sup>19</sup>, SSL<sup>20</sup>, JSON<sup>21</sup>.

---

<sup>5</sup> <https://www.w3schools.com>

<sup>6</sup> <https://www.ubuntu.com>

<sup>7</sup> <https://www.phpmyadmin.net/>

<sup>8</sup> <https://www.mysql.com>

<sup>9</sup> <https://www.w3schools.com/bootstrap4>

<sup>10</sup> <https://www.wampserver.com/>

<sup>11</sup> <https://getcomposer.org>

<sup>12</sup> <https://httpd.apache.org/>

<sup>13</sup> <https://code.visualstudio.com/docs>

<sup>14</sup> <https://www.virtualbox.org>

<sup>15</sup> <https://www.draw.io>

<sup>16</sup> <https://www.virtualbox.org> / <https://www.w3schools.com/html>

<sup>17</sup> <https://www.w3schools.com/css/>

<sup>18</sup> <https://jquery.com/>

<sup>19</sup> <https://www.w3.org/XML/>

<sup>20</sup> <https://www.ssl.com>

<sup>21</sup> <https://www.json.org>

### 3. Web Application developed

In this section we'll show the web apps developed to implement 3D Secure protocol.

#### 3.1. Banks web application:

To simulate a real bank, we have developed this web app to create customers, accounts, cards, transactions, account types, users....etc., we have explained it in brief by showing use cases, sequence diagram, class diagram, functionalities list and source code.

##### A) Use cases:

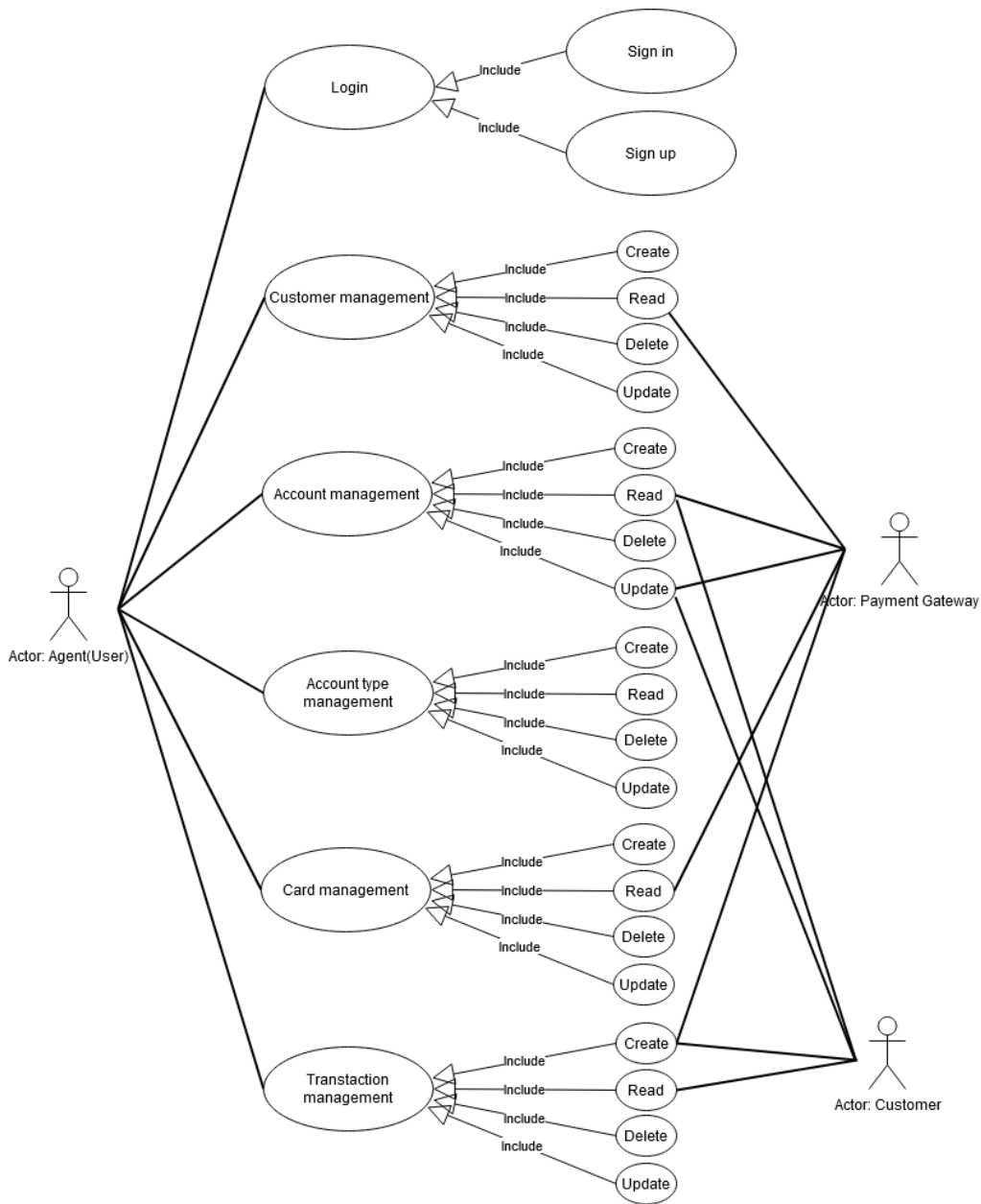
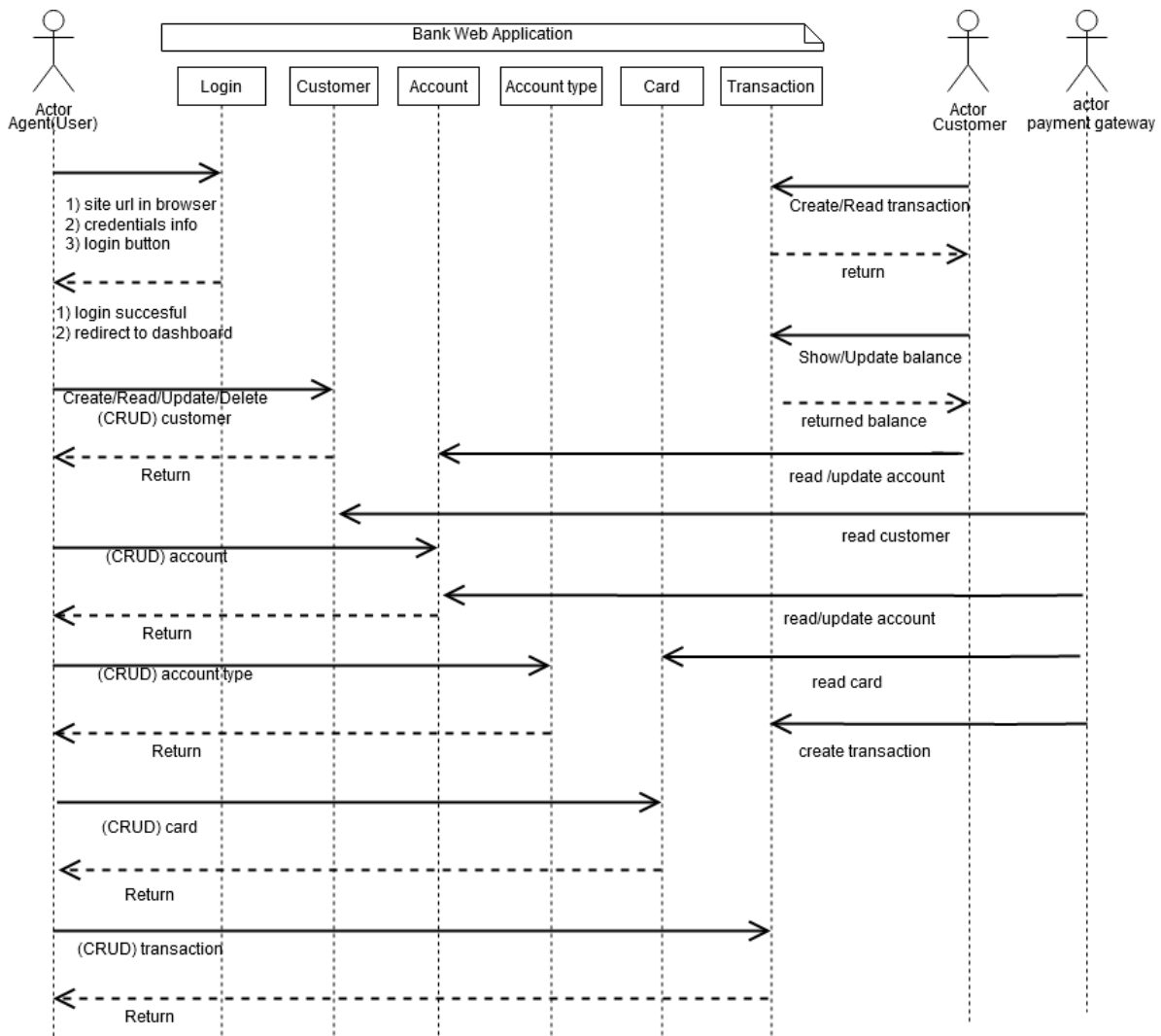


Figure 22: Use cases of Banking web app.

**B) Sequence diagram:**



**Figure 23: Sequence diagram of Banking web app.**

**C) Database diagram:**

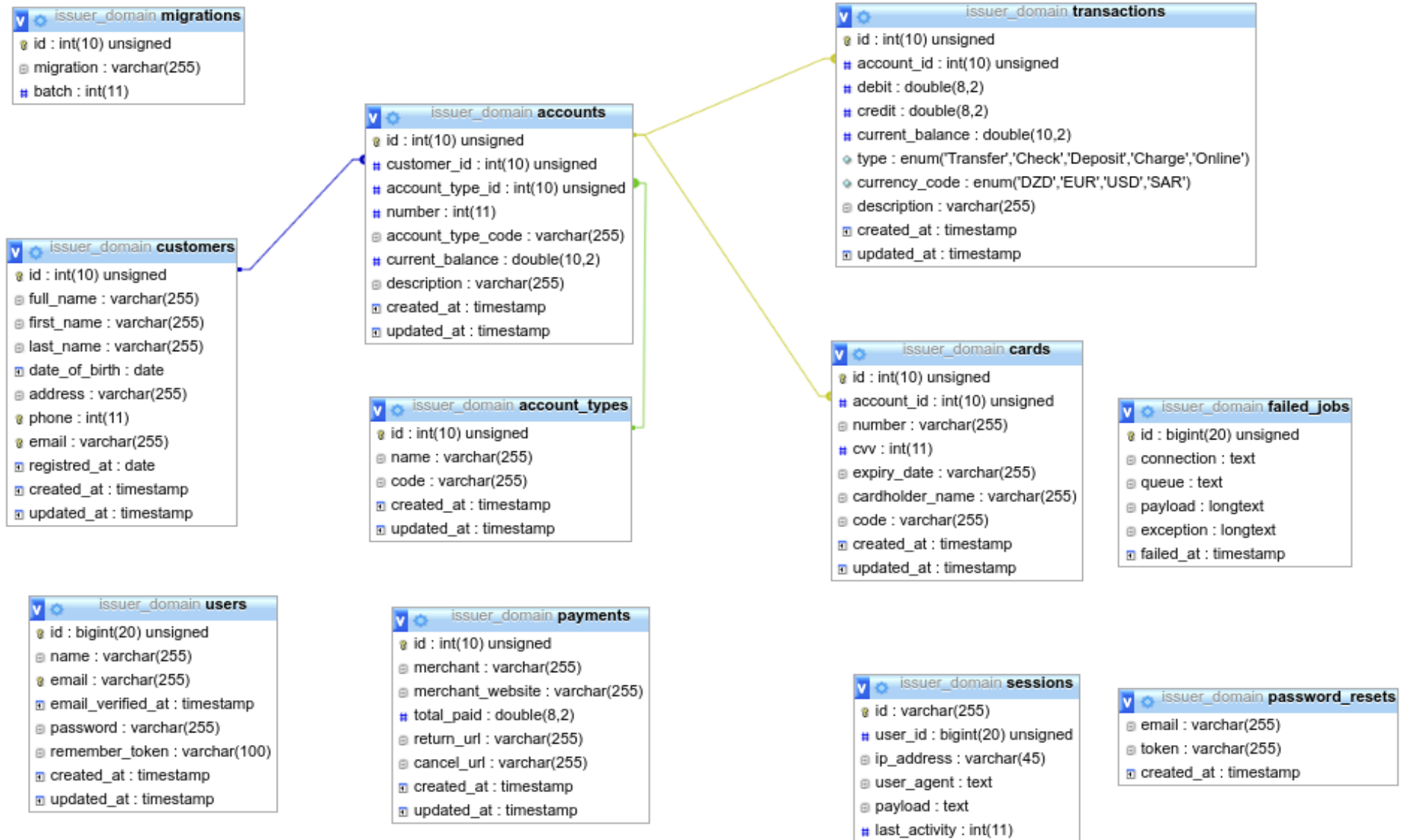


Figure 24: Database diagram of Banking web app.

**D) Features list:**

Feature	Operations
Analytics dashboard	Last operations, notifications,...
Customer management	Add, View, edit and delete
Account Management	Add, View, edit and delete
Account type management	Add, View, edit and delete
Transaction management	Add, View, edit and delete
Card management	Add, View, edit and delete
Settings/Administration	View, edit parameters
Mobile	Yes, Responsive
Notification system	yes , using emails

**Table 7: Features list of Banking web app.****E) Source code:**

We have used Laravel to develop this web application. Because the source code is very large, we have been uploading the entire project in GitHub repository, to download or take a look to this project; click on this link:

<https://github.com/NasrEddineDev/Master-Thesis-ImplementationOf3DSecureArchitecture.git>

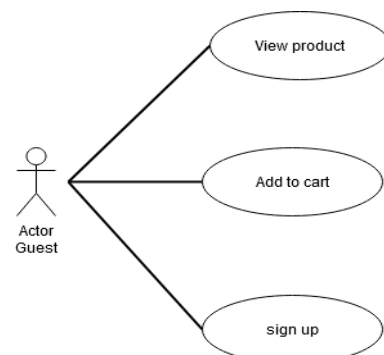
**3.2. Ecommerce web app:**

To verify our implementation of 3D Secure protocol, we have developed this ecommerce website to make online shopping and allows consumers to directly buy goods or services from a seller (Merchant) over the Internet using a web browser. ...

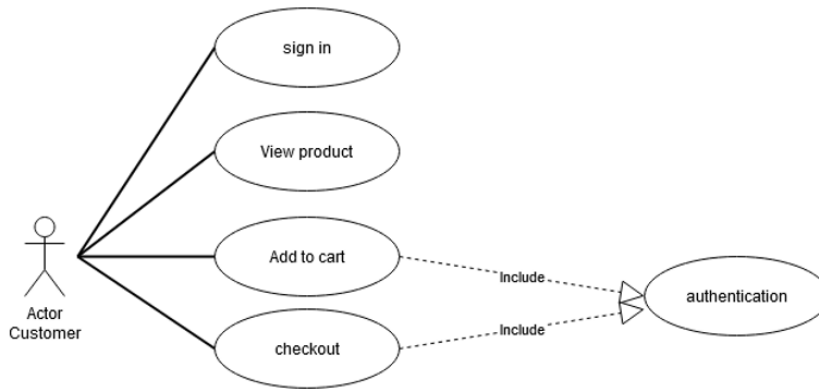
This web application contains two separate interfaces, the first one is an admin panel to control the online shop by the merchant, this include: customers, products, categories, employees, and users management...etc. and showing orders list, and order status list...etc. In this section and to explain well this web application, we will present the use cases, sequence diagram, class diagram, functionalities list and source code.

**A) Use case:**

- **Guest:**

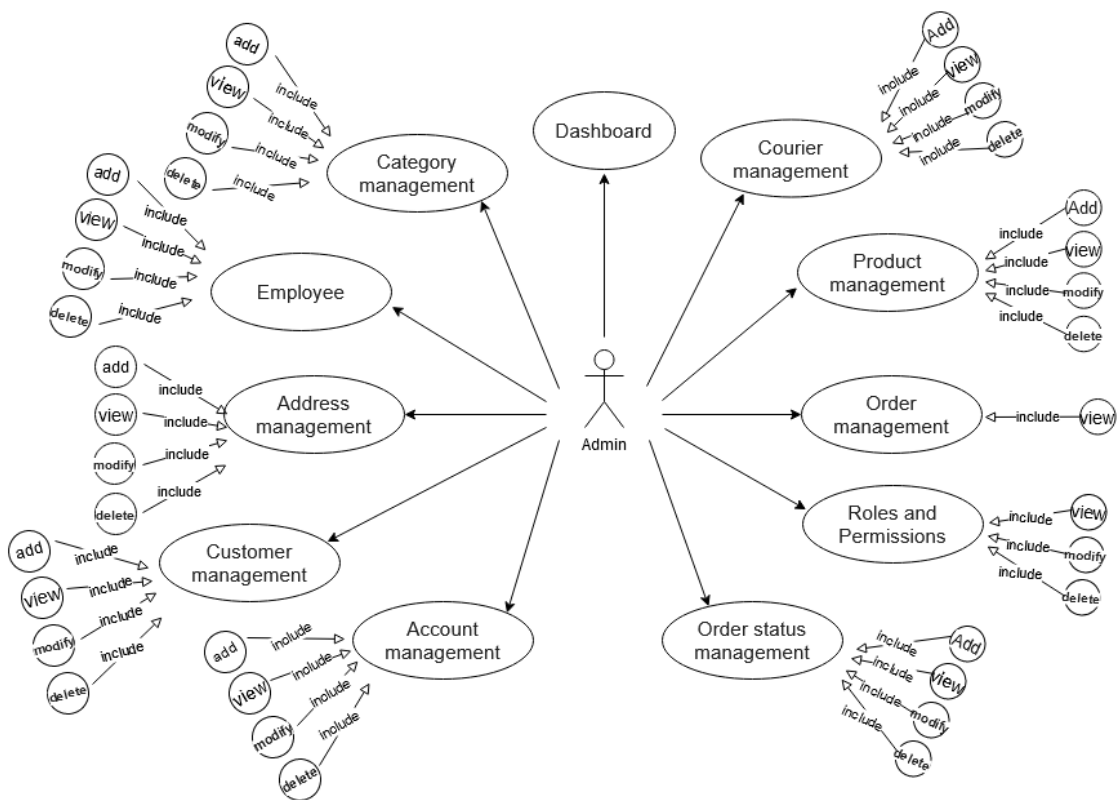
**Figure 25: Guest use cases of ecommerce web app**

• **Customer:**



**Figure 26: Customer use cases of ecommerce web app**

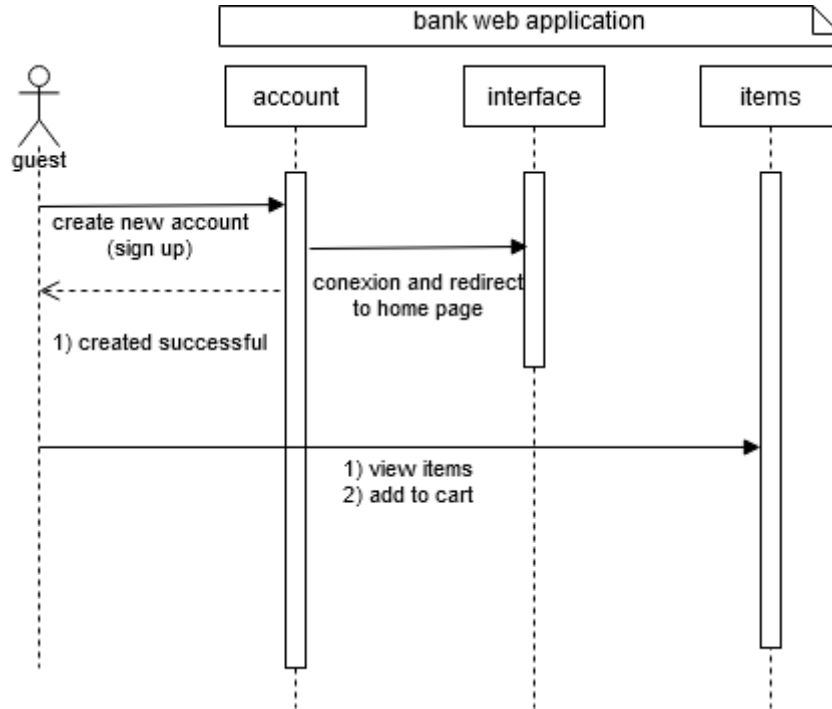
• **Admin:**



**Figure 27: Admin use cases of e-commerce web app.**

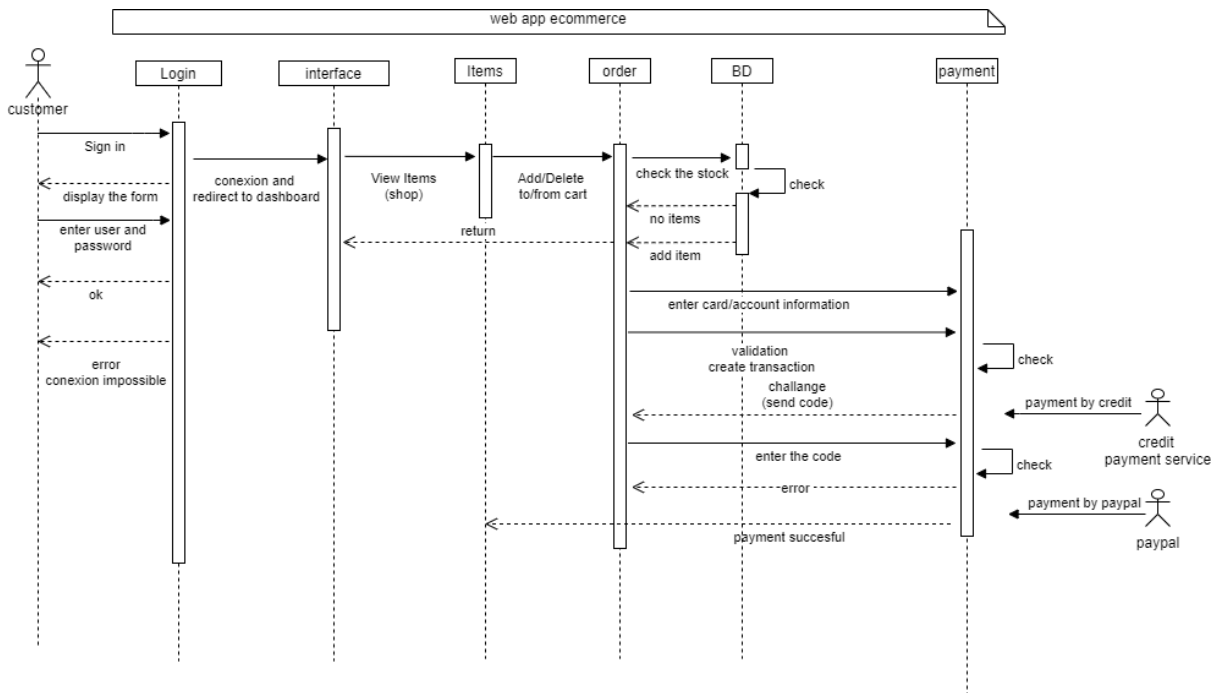
**B) Sequence diagram:**

- **Guest:**



**Figure 28: Guest sequence diagram of ecommerce web app**

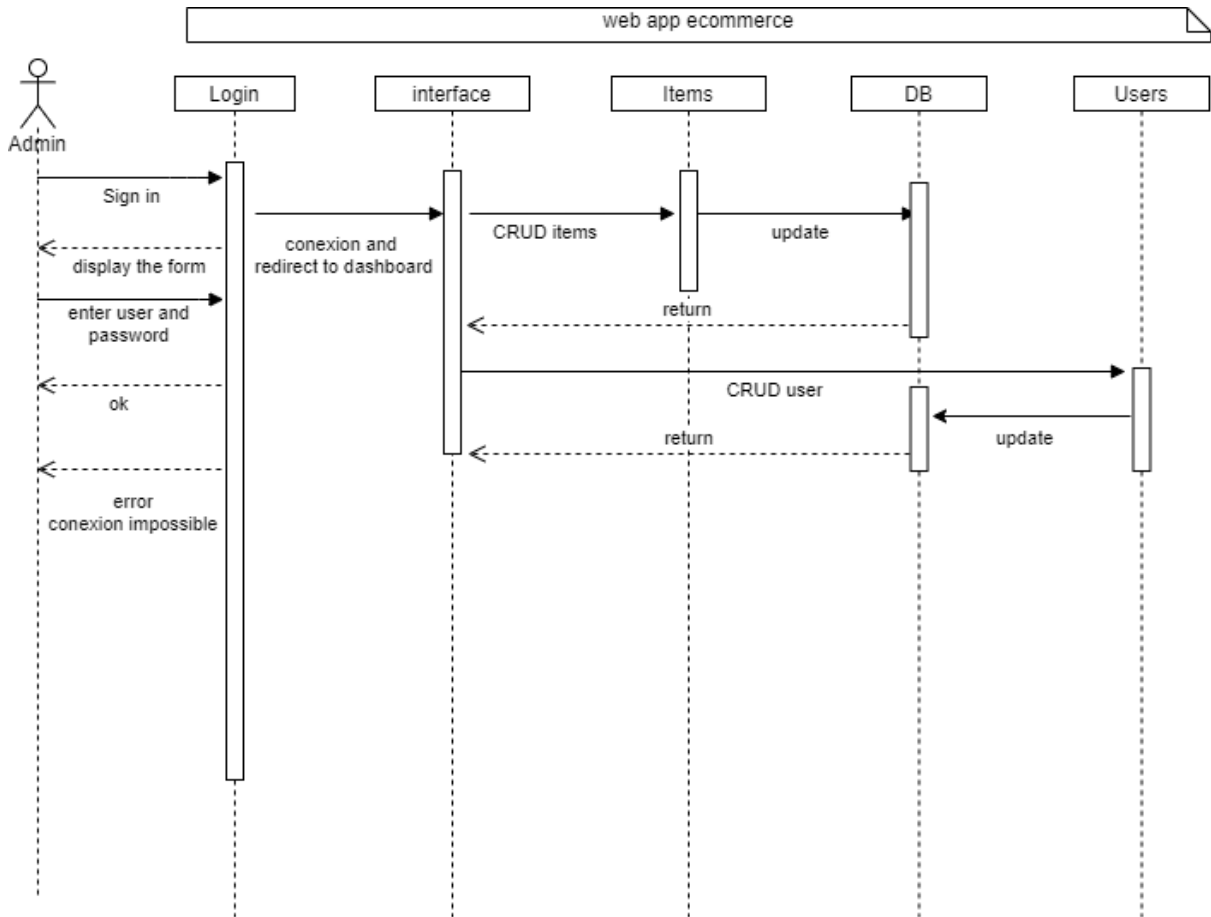
- **Customers:**



**Figure 29: Customer sequence diagram of ecommerce web app**

- **Admin:**

In this part, the term items has been used to refer to products, categories, and employees...etc.



**Figure 30: Admin sequence diagram of ecommerce web app.**



C) Database diagram:

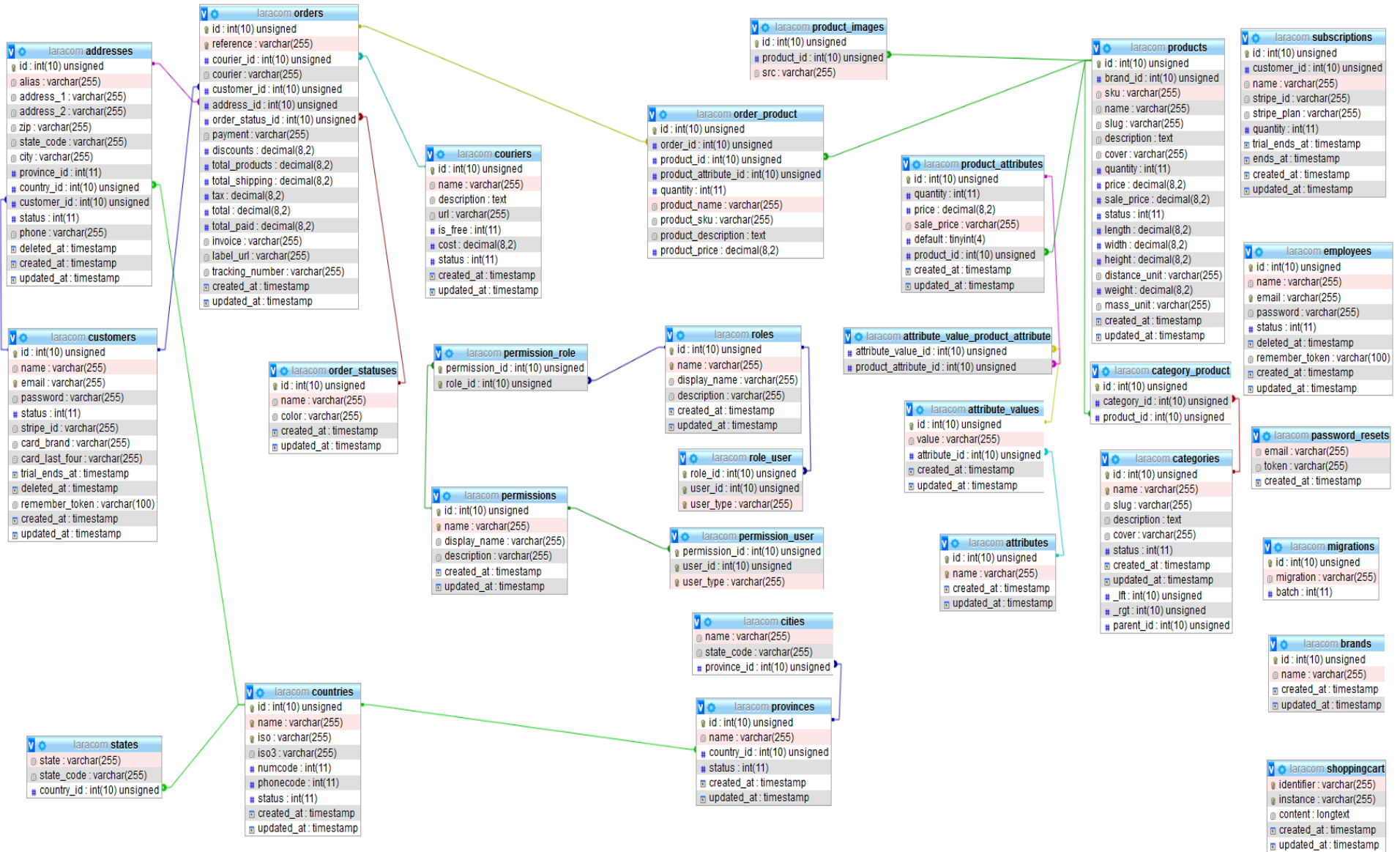


Figure 31: Database diagram of ecommerce web app.

**D) Features list:**

Sub web app	Feature	Operations
User	Account management	Add, Details, modify
	Product management	List, Details, Add to cart, Buy
	Address management	List, Add, modify and delete
	Order management	List
	Mobile	Yes, Responsive
	Search, Filters	Yes
Admin	Dashboard	
	Customer management	List, Add, Details, modify and delete
	Address management	List, Add, Details, modify and delete
	Account management	List, Add, Details, modify and delete
	Category management	List, Add, Details, modify and delete
	Product management	List, Add, Details, modify and delete
	Order management	List, Details
	Order status management	List, Add, modify and delete
	Courier management	List, Add, modify and delete
	Employee	List, Add, Details, modify and delete
	Roles and Permissions	List, modify and delete
	Mobile	Yes, Responsive

**Table 8: Features list of ecommerce web app.****E) Source code:**

In order to develop this web application (Hanouti), we have also used the awesome framework Laravel<sup>22</sup>. Because the source code is very large (more than 100 Mo!), we have been uploading the entire project in GitHub repository, to download or take a look to this project; click on this link :

<https://github.com/NasrEddineDev/Master-Thesis-ImplementationOf3DSecureArchitecture.git>

---

<sup>22</sup> If you have interested to develop a modern web app, please visit official website <https://laravel.com> to learn more about this wonderful and open source framework,

## 4. Source codes added to implement 3D Secure

In this part, we reference all our source codes added to implement 3D Secure architecture. We therefore believed that it was appropriate to divide this part to fore subpart. The first contains source code of payment gateway web app itself, while the second (MPI) contains source code inserted into the ecommerce web site to interact with the payment gateway, the two next contains source code developed into issuer and acquirer banks to interact with payment gateway or interoperability domain or customers, the last part contains source code developed for interoperability domain to interact between issuer and acquirer domains. To see these parts of source codes, please refer to the appendix B.

## 5. Results

In order to be able to understand well, and to explain well our work, we will take full web page screenshots of the 3D Secure protocol process, beginning from step 01 until coming to step 08. We will show a detailed demonstration starting from creating accounts until coming to the end of a successful online shopping process where we send payment receipts to the customer.

### 5.1. Table of parameters

#### a. Customers:

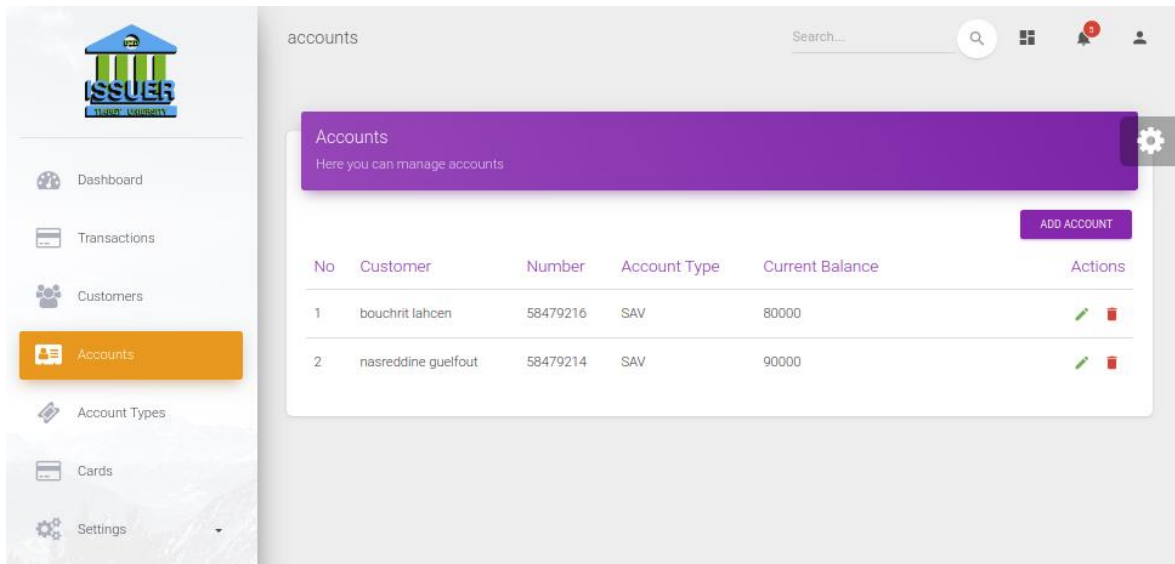
	Issuer bank		Acquirer bank
Customers Property	Customer 1	Customer 2	Customer 3
full_name	Bouchrit lahcen	guelfout nasreddine	aek mohamed
Date of birth	17/01/1992	1988-10-23	1992-07-15
address	Ain dheb tiaret	Ain mesbah tiaret	Tiaret
Email(@gmail.com)	lahcenbouchrit	mehamednasreddine	hanoutidzshopping
phone	676856785	6666666	2132345678

**Table 9: Customers parameters.**

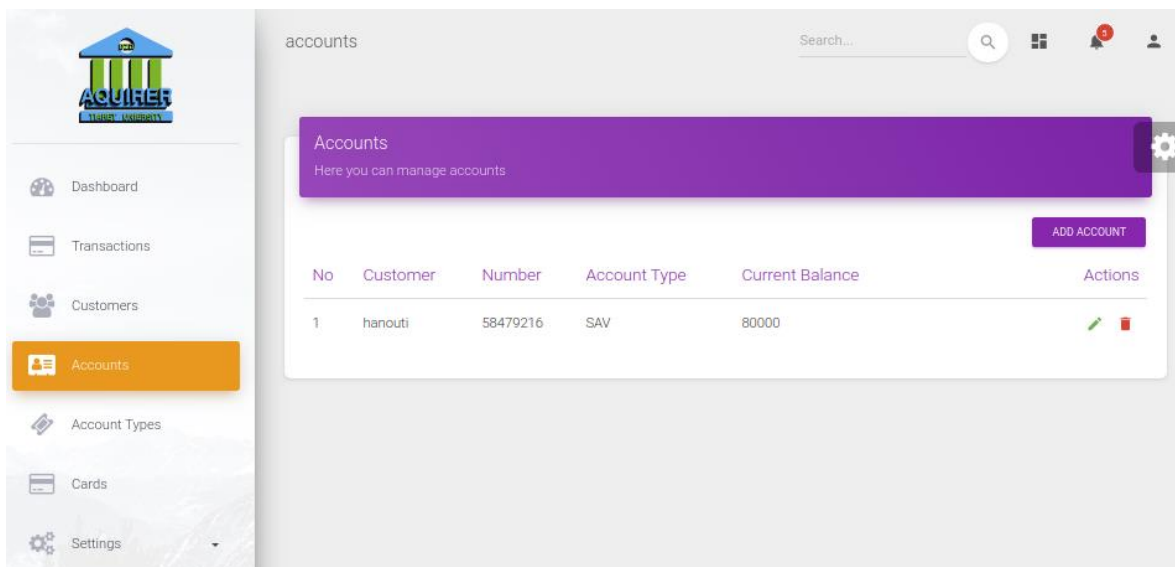
#### b. Account:

	Issuer bank		Acquirer bank
Customers Property	1	2	3
Number	58479216	58479214	58479243
Account Type	SAV	SAV	SAV
Balance	80000	90000	80000

**Table 10: Accounts parameters**



**Figure 32: Issuer accounts.**



**Figure 33: Acquirer accounts.**

**Cards:**

	Issuer bank		Acquirer bank
Customers	Customer 1	Customer 2	Customer 3
Property			
Account	58479216	58479214	
cardholder_name	lahcen bouchrit	guelfout nasreddine	
card_number	1111222233334444	1111222233334445	
CVV	123	321	
exp_date	01/22	02/22	

**Table 11: Cards parameters.**

## 5.2. Online Shopping Steps using 3D Architecture

### Step 01: Make online shopping by cardholder

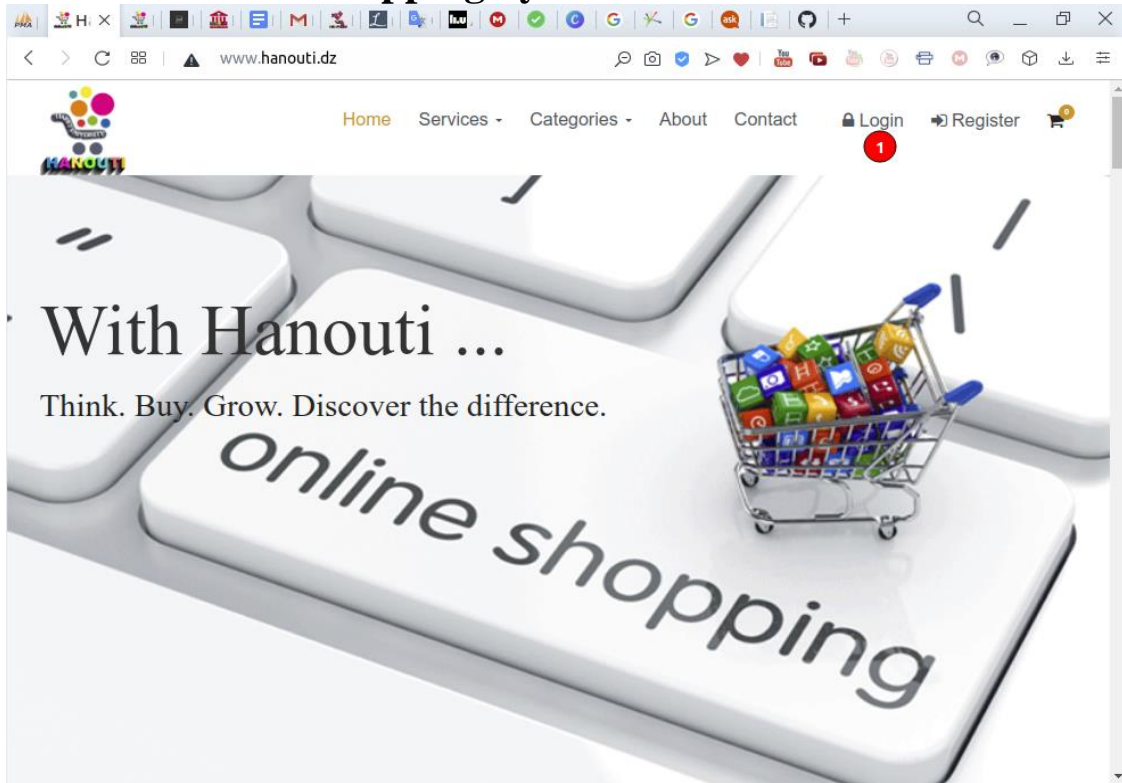


Figure 34: Hanouti web app home page.

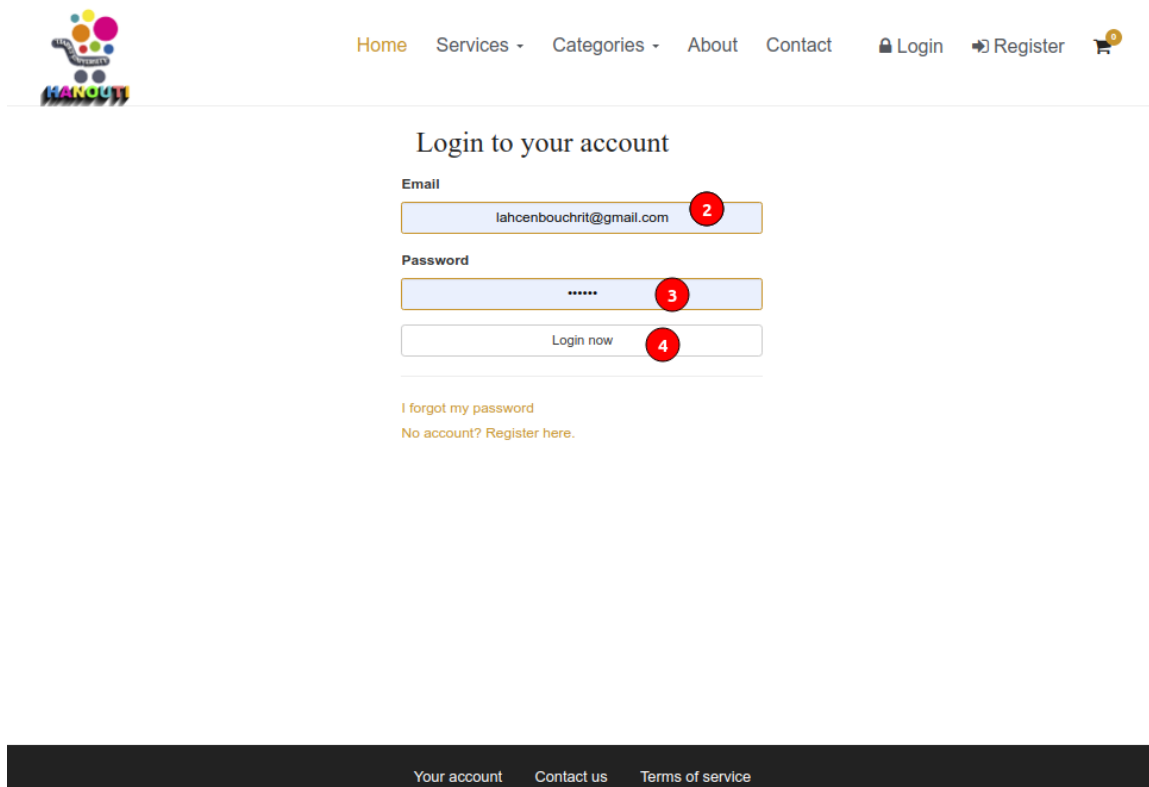
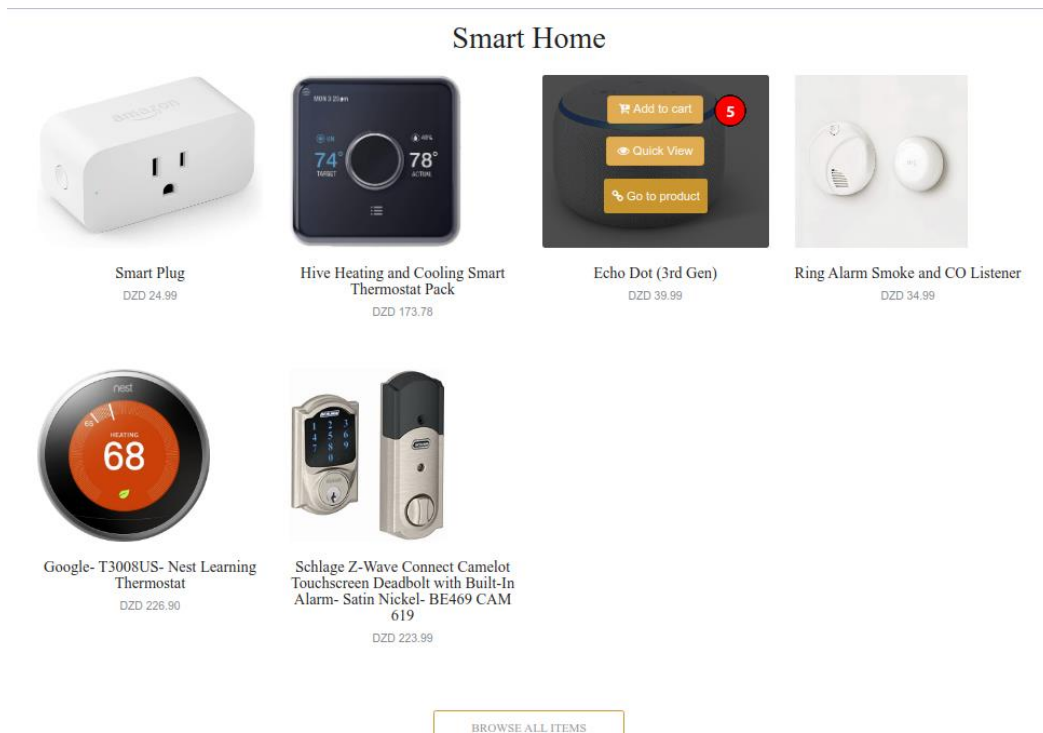
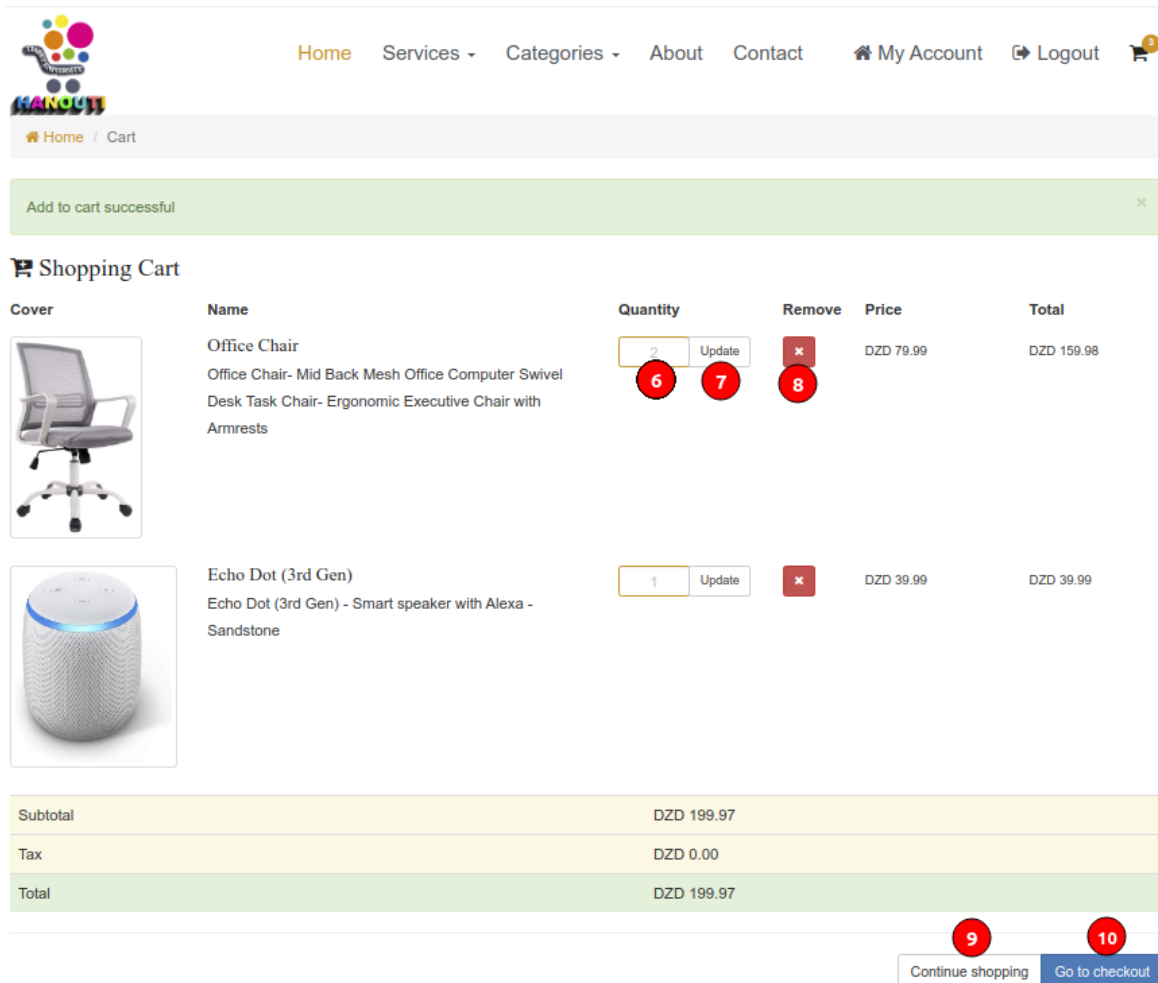


Figure 35: Hanouti login page





**Figure 36: Hanouti add products to cart.**



**Figure 37: Hanouti checkout page.**

Home Services Categories About Contact My Account Logout

Home / Shopping Cart

Cover	Name	Quantity	Remove	Price	Total
	Office Chair Office Chair- Mid Back Mesh Office Computer Swivel Desk Task Chair- Ergonomic Executive Chair with Armrests	2 Update	<input type="button" value="x"/>	DZD 0.00 DZD 159.98	DZD 79.99
	Echo Dot (3rd Gen) Echo Dot (3rd Gen) - Smart speaker with Alexa - Sandstone	1 Update	<input type="button" value="x"/>	DZD 0.00 DZD 39.99	DZD 39.99

Addresses

Alias	Address	Billing Address	Delivery Address
TIARET	ain dhab, TIARET, Algeria Adak AK UNITED STATES OF AMERICA 4545413	<input type="checkbox"/>	<input checked="" type="checkbox"/> Same as billing

Payment

Name	Description	Choose payment
Pgw-pfe3ds	pgw-pfe3ds - Safe, Secured and Easy to pay online!	<input type="button" value="Pay with Pgw-pfe3ds"/>
Paypal	PayPal - Safe, Secured and Easy to pay online!	<input type="button" value="Pay with Paypal"/>
Stripe	The new standard in online payments	<input type="button" value="Pay with Stripe"/>
Bank Transfer	Online / Offline Bank fund transfer	<input type="button" value="Pay with Bank Transfer"/>

Figure 38: Hanouti, choose the payment method.



## Step 02: Submit card information to Payment Gateway

PGW-PFE3DS Payment Gateway

Merchant information

Merchant : hanouti      Amount : 199.97 DA  
Web site : http://hanouti.dz

Payment Details

Card Owner : bouchrit lahcen **12**

Card Number : 1111-2222-3333-4445 **13**

EXPIRY DATE : 02/22 **14**      CODE : ... **15**

**16** SUBMIT      **17** CANCEL

© 2020 , made by University of tiaret

Figure 39: PGW-PFE3DS, information card page.

## Step 03: Send code

PGW-PFE3DS Payment Gateway

Transaction confirmation

A unique code will be sent to this email: lah\*\*\*\*\*com

Merchant : hanouti  
Amount : 199.97 DA  
Date : 2020-10-06 21:44:30  
Card number : 111\*\*\*\*\*444

**18** SEND

© 2020 , made by University of tiaret

Figure 40: PGW-PFE3DS, send verification code.



## Step 04: verification email

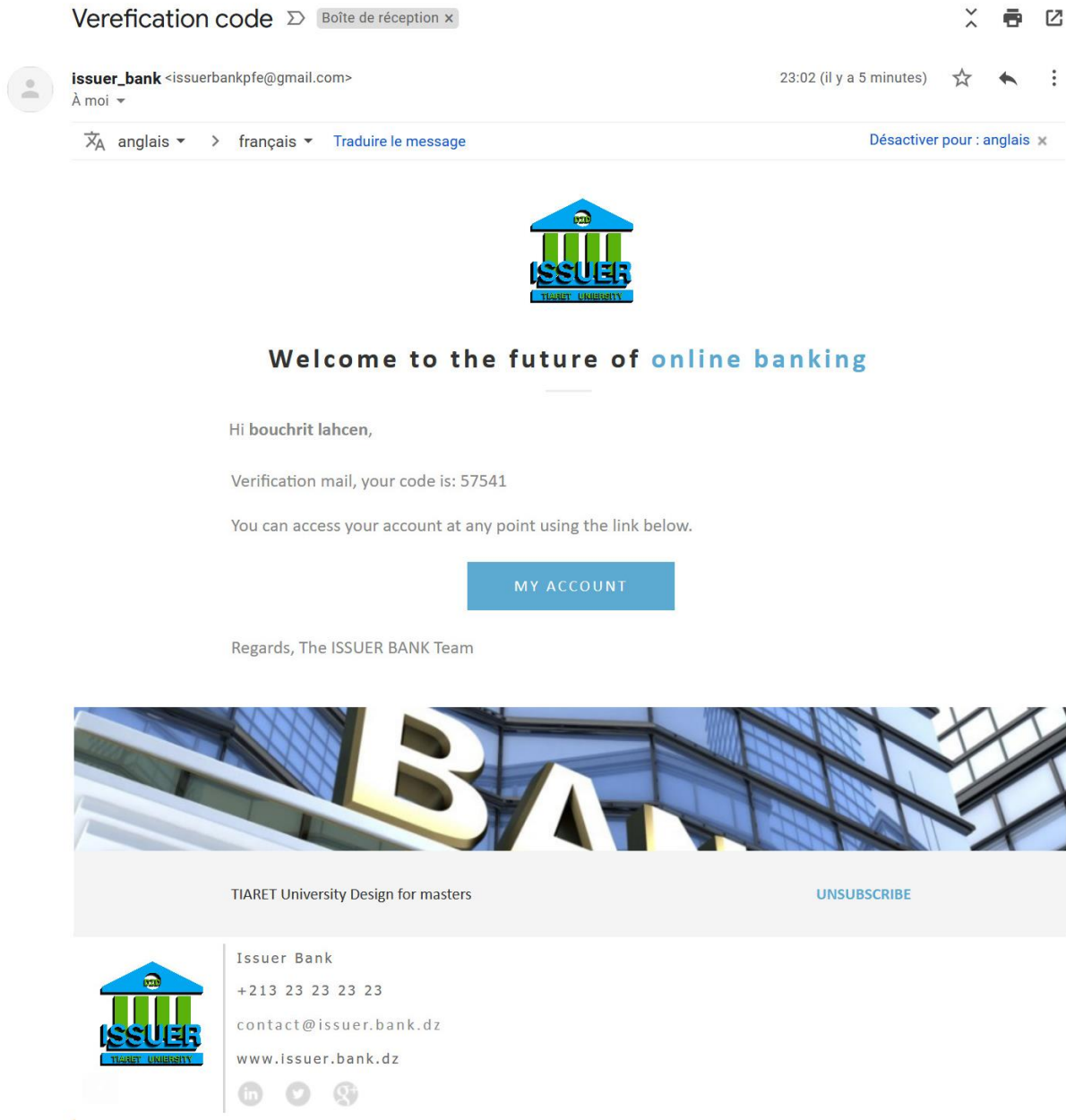


Figure 41: Issuer send verification email

## Step 05: type code



Figure 42: PGW-PFE3DS, Enter the code.

## Step 06: Make transaction

### a. Customer account (issuer bank)

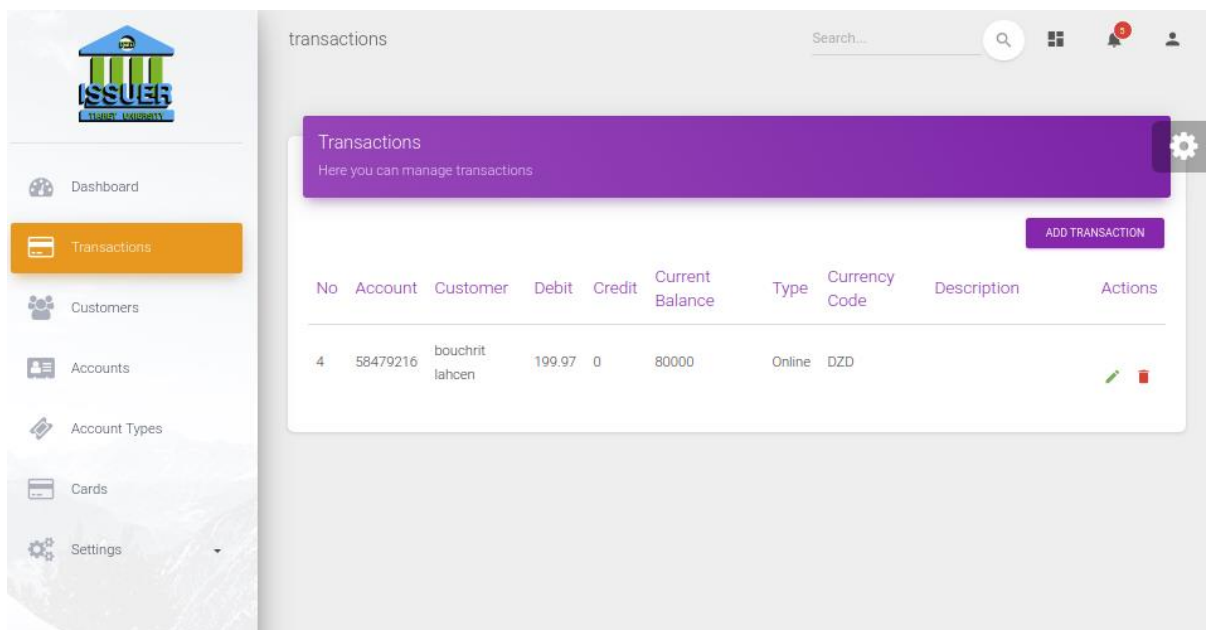
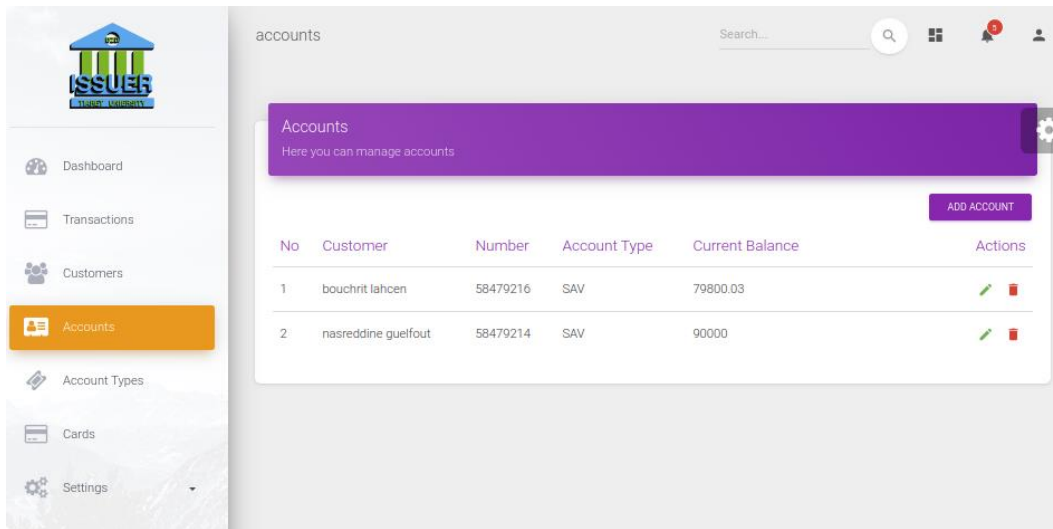
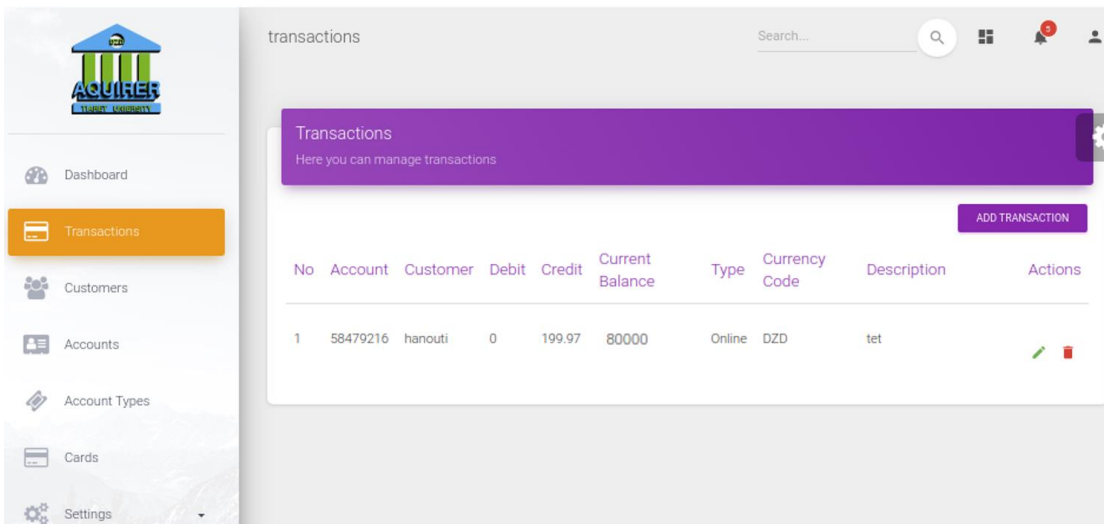


Figure 43: Issuer transaction



**Figure 44: Issuer account balance**

**b. Merchant account (acquirer bank)**



**Figure 45: Acquirer transaction**

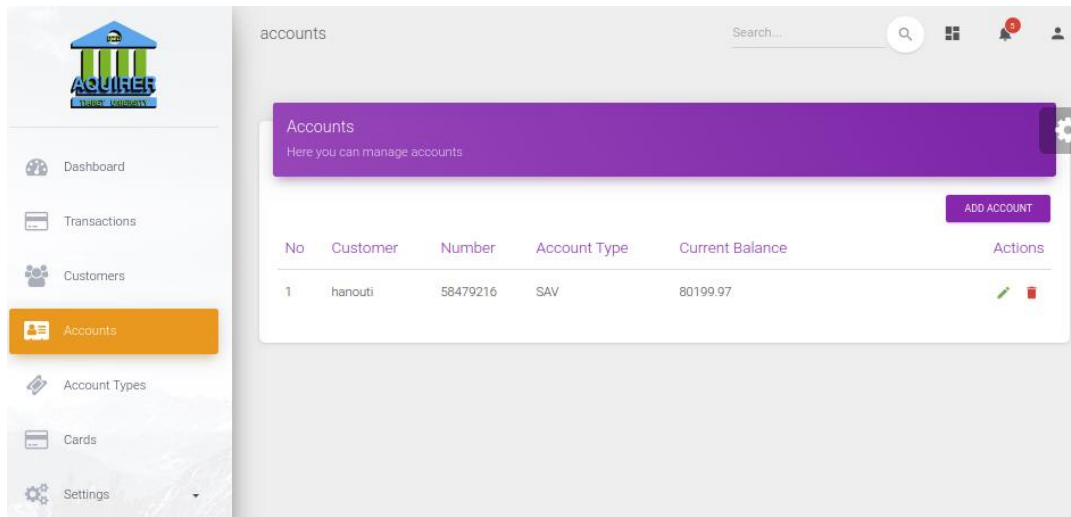


Figure 46: Acquirer account.

Step 07:payment receipt




Figure 47: Hanouti payment receipt.

## Step 08: Send payment receipt via email

New order > Boîte de réception x

hanouti <hanoutidzshopping@gmail.com> 23:06 (il y a 3 minutes) ☆ ↶ ⋮  
À moi ▾

anglais > français Traduire le message Désactiver pour : anglais x



**Hanouti**  
Ibn-Khaldoun University, Tiaret, Algeria.  
+213(0)46 20 88 49  
[contact@hanouti.dz](mailto:contact@hanouti.dz)


INVOICE TO:  
**bouchrit lahcen**  
ain dhab, TIARET, Algeria  
[lahcenbouchrit@gmail.com](mailto:lahcenbouchrit@gmail.com)

**INVOICE 2548**  
Date of Invoice: 2020-10-06  
Due Date: 2020-10-06


#	NAME	DESCRIPTION	UNIT PRICE	QUANTITY	TOTAL
01	Office Chair	Office Chair- Mid Back Mesh Office Computer Swivel Desk Task Chair- Ergonomic Executive Chair with Armrests	79.99	2	DZD 159.98
01	Echo Dot (3rd Gen)	Echo Dot (3rd Gen) - Smart speaker with Alexa - Sandstone	39.99	1	DZD 39.99
SUBTOTAL					DZD 199.97
TAX					0.00
TOTAL					DZD 199.97

Thank you!  
NOTICE:  
A refund will be given on undelivered products after 3 days.

Invoice was created on a computer and is valid without the signature and seal.



TIARET University Design for masters UNSUBSCRIBE



Hanouti online shop  
+213 23 23 23 23  
[contact@hanouti.dz](mailto:contact@hanouti.dz)  
[www.hanouti.dz](http://www.hanouti.dz)

[in](#) [t](#) [s](#)

Figure 48: Hanouti, send payment receipt via email.

## **6. Conclusion**

In this chapter, we have described the material and software used and their source to implement the 3D Secure architecture, we have also written our source code integrated in the merchant website and payment gateway, issuer, acquirer, and interoperability domains. We have also presented for each web application the use cases, sequence diagram, class diagram, feature list, and a brief description of the source code. Finally, we have shown a detailed demonstration of a successful online shopping process.

# Conclusion

Credit cards and online payment are now becoming a part of our daily life due to its benefits. Today, the online shopping transactions continues to grow rapidly on a regular basis and the number of its users are increasing rapidly in relation to the tremendous changes that continue to occur within the technology. Therefore, there will be an increase in the number and types of attacks against the security of online payment systems. Consumers may be at the risk of losing their money and personal data. Thus, it is very important to make the Internet safe for buying and selling the products online.

3D Secure is one of the most widely used protocols to secure online payments. In this project, we have a detailed review of how to implement the 3D secure protocol according to the EMV standard and adapt it for the ecommerce market in Algeria, and also how to make the double authentication using SMS and email via one-time password (OTP), or other mechanisms.

This document is divided into three main sections. The first section gives a global overview of ecommerce and different parts of E-payment security (the state of the art).

The second section examines 3D Secure protocol components and messages, a general activity diagram of the protocol is outlined in this section, a detailed and explained steps of this protocol, and finally we will show the 3D Secure pseudocode.

In the last section, we have described how to implement 3D Secure protocol step by step and the different technologies that we have used, including web application development and the source code of the protocol. Finally, we have a demonstration to prove the success of our implementation.

However, further work needs to be done to enhance the security against the growth of online transactions frauds and attacks. We propose that further research should be undertaken in the following areas:

- How to automate the second authentication (without OTP and user interaction).
- How to identify the cardholder without the duplicate authentication using AI (computer vision).

# Bibliography

- [1] Y. A. Nanehkaran, "An Introduction To Electronic Commerce," *International Journal of Scientific & Technology Research*, vol. 2, no. 4, pp. 190-193, APRIL 2013.
- [2] K. Moore, "Ecommerce," 13 February 2018. [Online]. Available: <https://www.bigecommerce.com>.
- [3] C. M. M. A SENGUPTA, "e-Commerce security, A life cycle approach," S<sup>adha</sup>na, India, April/June 2005.
- [4] P. D. M. Kütz, Introduction to E-Commerce Combining Business and Information Technology, bookboon.com The eBook company, 2016.
- [5] R. S. Network Working Group, "RFC 4949 Internet Security Glossary," August 2007.
- [6] J. Y. HUA JIANG, "On-line Payment and Security of E-commerce," in *WSEAS International Conference on Computer Engineering and Applications*, Gold Coast, Australia, January 17-19, 2007.
- [7] A. Koponen, "E-COMMERCE, ELECTRONIC PAYMENTS," Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory.
- [8] A. Bouch, "3-D Secure: A critical review of 3-D Secure and its effectiveness in preventing card not present fraud," University of London., London, March 2011.
- [9] EMVCo, EMV 3-D Secure Protocol and Core Functions Specification v2.2.0, 2018.



# Appendix

## Appendix A: Message Format

This annex provides the EMV 3D Secure data elements and field names by Message Type [9].

### A.1. AReq Message Data Elements

Data Element	Field Name
3DS Method Completion Indicator	threeDSCompInd
3DS Requestor Authentication Indicator	threeDSRequestorAuthenticationInd
3DS Requestor Authentication Information	threeDSRequestorAuthenticationInfo
3DS Requestor Authentication Method Verification Indicator	threeDSReqAuthMethodInd
3DS Requestor Challenge Indicator	threeDSRequestorChallengeInd
3DS Requestor Decoupled Max Time	threeDSRequestorDecMaxTime
3DS Requestor Decoupled Request Indicator	threeDSRequestorDecReqInd
3DS Requestor ID	threeDSRequestorID
3DS Requestor Name	threeDSRequestorName
3DS Requestor Prior Transaction Authentication Information	threeDSRequestorPriorAuthenticationInfo
3DS Requestor URL	threeDSRequestorURL
3DS Server Reference Number	threeDSServerRefNumber
3DS Server Operator ID	threeDSServerOperatorID
3DS Server Transaction ID	threeDSServerTransID
3DS Server URL	threeDSServerURL
3RI Indicator	threeRIInd
Account Type	acctType
Acquirer BIN	acquirerBIN
Acquirer Merchant ID	acquirerMerchantID
Address Match Indicator	addrMatch
Broadcast Information	broadInfo
Browser Accept Headers	browserAcceptHeader
Browser IP Address	browserIP
Browser Java Enabled	browserJavaEnabled
Browser JavaScript Enabled	browserJavascriptEnabled
Browser Language	browserLanguage
Browser Screen Color Depth	browserColorDepth
Browser Screen Height	browserScreenHeight
Browser Screen Width	browserScreenWidth
Browser Time Zone	browserTZ
Browser User-Agent	browserUserAgent
Card/Token Expiry Date	cardExpiryDate
Cardholder Account Information	acctInfo
Cardholder Account Number	acctNumber
Cardholder Account Identifier	acctID
Cardholder Billing Address City	billAddrCity
Cardholder Billing Address Country	billAddrCountry
Cardholder Billing Address Line 1	billAddrLine1
Cardholder Billing Address Line 2	billAddrLine2
Cardholder Billing Address Line 3	billAddrLine3
Cardholder Billing Address Postal Code	billAddrPostCode
Cardholder Billing Address State	billAddrState
Cardholder Email Address	email

Cardholder Home Phone Number	homePhone
Cardholder Mobile Phone Number	mobilePhone
Cardholder Name	cardholderName
Cardholder Shipping Address City	shipAddrCity
Cardholder Shipping Address Country	shipAddrCountry
Cardholder Shipping Address Line 1	shipAddrLine1
Cardholder Shipping Address Line 2	shipAddrLine2
Cardholder Shipping Address Line 3	shipAddrLine3
Cardholder Shipping Address Postal Code	shipAddrPostCode
Cardholder Shipping Address State	shipAddrState
Cardholder Work Phone Number	workPhone
Device Channel	deviceChannel
Device Information	deviceInfo
Device Rendering Options Supported	deviceRenderOptions
DS Reference Number	dsReferenceNumber
DS Transaction ID	dsTransID
DS URL	dsURL
EMV Payment Token Indicator	payTokenInd
EMV Payment Token Source	payTokenSource
Instalment Payment Data	purchaseInstalData
Merchant Category Code	mcc
Merchant Country Code	merchantCountryCode
Merchant Name	merchantName
Merchant Risk Indicator	merchantRiskIndicator
Message Category	messageCategory
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Notification URL	notificationURL
Purchase Amount	purchaseAmount
Purchase Currency	purchaseCurrency
Purchase Currency Exponent	purchaseExponent
Purchase Date & Time	purchaseDate
Recurring Expiry	recurringExpiry
Recurring Frequency	recurringFrequency
SDK App ID	sdkAppID
SDK Encrypted Data	sdkEncData
SDK Ephemeral Public Key (Qc)	sdkEphemPubKey
SDK Maximum Timeout	sdkMaxTimeout
SDK Reference Number	sdkReferenceNumber
SDK Transaction ID	sdkTransID
Transaction Type	transType
Whitelist Status	whiteListStatus
Whitelist Status Source	whiteListStatusSource

Table 12: AReq Data Elements.

## A.2. ARes Message Data Elements

Data Element	Field Name
3DS Server Transaction ID	threeDSserverTransID
ACS Challenge Mandated Indicator	acsChallengeMandated
ACS Decoupled Confirmation Indicator	acsDecConInd
ACS Operator ID	acsOperatorID
ACS Reference Number	acsReferenceNumber
ACS Rendering Type	acsRenderingType
ACS Signed Content	acsSignedContent
ACS Transaction ID	acsTransID
ACS URL	acsURL
Authentication Type	authenticationType
Authentication Value	authenticationValue

Broadcast Information	broadInfo
Cardholder Information Text	cardholderInfo
DS Reference Number	dsReferenceNumber
DS Transaction ID	dsTransID
Electronic Commerce Indicator	eci
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
SDK Transaction ID	sdkTransID
Transaction Status	transStatus
Transaction Status Reason	transStatusReason
Whitelist Status	whiteListStatus
Whitelist Status Source	whiteListStatusSource

**Table 13: ARes Data Elements.****A.3. CReq Message Data Elements**

Data Element	Field Name
3DS Requestor App URL	threeDSRequestorAppURL
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
Challenge Cancelation Indicator	challengeCancel
Challenge Data Entry	challengeDataEntry
Challenge HTML Data Entry	challengeHTMLDataEntry
Challenge No Entry	challengeNoEntry
Challenge Window Size	challengeWindowSize
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
OOB Continuation Indicator	oobContinue
Resend Challenge Information Code	resendChallenge
SDK Transaction ID	sdkTransID
SDK Counter SDK to ACS	sdkCounterStoA
Whitelisting Data Entry	whitelistingDataEntry

**Table 14: CReq Data Elements.****A.4. CRes Message Data Elements**

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Counter ACS to SDK	acsCounterAtoS
ACS Transaction ID	acsTransID
ACS HTML	acsHTML
ACS UI Type	acsUiType
Challenge Completion Indicator	challengeCompletionInd
Challenge Information Header	challengeInfoHeader
Challenge Information Label	challengeInfoLabel
Challenge Information Text	challengeInfoText
Challenge Information Text Indicator	challengeInfoTextIndicator
Challenge Selection Information	challengeSelectInfo
Expandable Information Label	expandInfoLabel
Expandable Information Text	expandInfoText
Issuer Image	issuerImage
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
OOB App URL	oobAppURL
OOB App Label	oobAppLabel
OOB Continuation Label	oobContinueLabel

Payment System Image	psImage
Resend Information Label	resendInformationLabel
SDK Transaction ID	sdkTransID
Submit Authentication Label	submitAuthenticationLabel
Whitelisting Information Text	whitelistingInfoText
Why Information Label	whyInfoLabel
Why Information Text	whyInfoText

**Table 15: CRes Data Elements.****A.5. Final CRes Message Data Elements**

Data Element	Field Description
3DS Server Transaction ID	threeDSSTransID
ACS Counter ACS to SDK	acsCounterAtoS
ACS Transaction ID	acsTransID
Challenge Completion Indicator	challengeCompletionInd
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
SDK Transaction ID	sdkTransID
Transaction Status	transStatus

**Table 16: Final CRes Data Elements.****A.6. PReq Message Data Elements**

Data Element	Field Name
3DS Server Reference Number	threeDSSTransRefNumber
3DS Server Operator ID	threeDSSTransOperatorID
3DS Server Transaction ID	threeDSSTransID
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Serial Number	serialNum

**Table 17: PReq Data Elements.****A.7. PRes Message Data Elements**

Data Element	Field Name
3DS Server Transaction ID	threeDSSTransID
Card Range Data	cardRangeData
DS End Protocol Version	dsEndProtocolVersion
DS Start Protocol Version	dsStartProtocolVersion
DS Transaction ID	dsTransID
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Serial Number	serialNum

**Table 18: PRes Data Elements.****A.8. RReq Message Data Elements**

Data Element	Field Name
3DS Server Transaction ID	threeDSSTransID
ACS Transaction ID	acsTransID
ACS Rendering Type	acsRenderingType
Authentication Method	authenticationMethod
Authentication Type	authenticationType
Authentication Value	authenticationValue

Challenge Cancellation Indicator	challengeCancel
DS Transaction ID	dsTransID
Electronic Commerce Indicator	eci
Interaction Counter	interactionCounter
Message Category	messageCategory
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
SDK Transaction ID	sdkTransID
Transaction Status	transStatus
Transaction Status Reason	transStatusReason
Whitelist Status	whiteListStatus
Whitelist Status Source	whiteListStatusSource

**Table 19: RReq Data Elements.****A.9. RRes Message Data Elements**

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
DS Transaction ID	dsTransID
Message Extension	messageExtension
Message Type	messageType
Message Version Number	messageVersion
Results Message Status	resultsStatus
SDK Transaction ID	sdkTransID

**Table 20: RRes Data Elements.****A.10. Error Messages Data Elements**

Data Element	Field Name
3DS Server Transaction ID	threeDSServerTransID
ACS Transaction ID	acsTransID
DS Transaction ID	dsTransID
Error Code	errorCode
Error Component	errorComponent
Error Description	errorDescription
Error Detail	errorDetail
Error Message Type	errorMessageType
Message Type	messageType
Message Version Number	messageVersion
SDK Transaction ID	sdkTransID

**Table 21: Error Message Data Elements.**

## Appendix B: 3D Secure Implementation (Source codes)

### B.1. Payment Gateway

#### a. Payment gateway library

```
<?php
namespace App\PGW;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Http;

trait PgwMethods
{
    protected function GetLastPaymentInfo()
    {
        return DB::table('payments')->latest()->first();
    }
    // Send Authentication Request message to $url
    protected function SendAReq($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
    // Send Challenge Request message to $url
    protected function SendCReq($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
    // Send Email Request to the $url of issuer
    protected function SendEmail($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
    // Send Code Validation Request to the issuer $url
    protected function ValidateCode($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
    // Send Payment Execution Request to the acquirer $url
    protected function ExecutePayment($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
}
```

#### b. API Methods

```
public function init(Request $request){
    $payment = new Payment([
        'merchant' => $request->merchant,
        'merchant_website' => $request->merchant_website,
        'total_paid' => $request->total_paid,
        'return_url' => $request->return_url,
        'cancel_url' => $request->cancel_url,
        'request_data' => json_encode($request->all())
    ]);
    $payment->save();
    return response()->json([
        'message' => '3DS protocol initiated successfully'
    ]);
}
```

```

        , 'content' => $request->merchant
    ], 200);
}

```

### c. Controller

```

public function index()
{
    $payment = $this->GetLastPaymentInfo();
    return view('pgw-pfe3ds.index', compact('payment'));
}
public function pay(Request $request)
{
    $payment = $this->GetLastPaymentInfo();
    $this->cardNumber = str_replace('-', '', $request->cardNumber);
    $payment->cardNumber = str_replace('-', '', $request->cardNumber);
    // sending AReq
    $messageAReq = [
        'threeDSRequestorID' => '',
        'threeDSRequestorName' => '',
        'threeDSRequestorURL' => '',
        'threeDSServerRefNumber' => '',
        'threeDSServerURL' => 'www.acquirer.bank.dz',
        'cardholderName' => $request->cardholderName,
        'acctNumber' => str_replace('-', '', $request->cardNumber),
        'cardExpiryDate' => $request->cardExpiryDate,
        'cvCode' => $request->cvCode, // remove or not ?
        'purchaseAmount' => $payment->total_paid,
        'acctType' => '',
        'acquirerBIN' => '',
        'acquirerMerchantID' => '',
        'email' => '',
        'mobilePhone' => '',
        'dsURL' =>
            'www.interbank.network.dz/api/interoperability/receiveareq',
        'merchantName' => $request->merchant,
        'purchaseAmount' => $request->total_paid,
        'purchaseDate' => now(),
        'acctID' => ''
    ];
    $acquirerURL = $messageAReq["threeDSServerURL"].
        '/api/acquirer/receiveareq';
    $response = $this->SendAReq($acquirerURL, $messageAReq);
    // Receiving ARes
    $messageARes = json_decode((string)$response->getBody());
    if (!empty($messageARes)) {
        if ($messageARes->statusCode == "401") {
            return redirect()->back()->withInput()
                ->withErrors([$messageARes->field => $messageARes->message]);
        }
        $payment->emailHiden = substr_replace($messageARes->email,
            '*****', 3, strlen($messageARes->email)-6);
        $payment->email = $messageARes->email;
        $payment->acsURL = $messageARes->acsURL;
        $payment->cardNumberHiden = substr_replace($payment->cardNumber,
            '*****', 3, strlen($payment->cardNumber)-6);
    }
    else {
        return redirect()->back()->withInput()
            ->withErrors(["cardExpiryDate" =>
                "There is an error on verifying card"]);
    }
    return view('pgw-pfe3ds.confirmation', compact('payment'));
}
public function sendCode(Request $request)

```

```

{
    // send CReq
    $result = $this->SendEmail($request->acsURL.'/api/issuer/sendemail',
        ['cardNumber' => $request->cardNumber,
        'email' => $request->email]);
    $response = json_decode((string)$result->getBody());
    return response()->json($response, 200);
}
public function validation(Request $request)
{
    $payment = $this->GetLastPaymentInfo();
    $response = $this->ValidateCode($request->acsURL .
        '/api/issuer/verifycode',
        ['code' => $request->code,
        'cardNumber' => $request->cardNumber,
        'amount' => $payment->total_paid,
        'merchant' => $payment->merchant]);
    $message = json_decode((string)$response->getBody());
    if (!empty($message) && $message->statusCode == '200'){
        // execute the transaction bank to bank
        $response = $this->ExecutePayment(
            'http://www.hanouti.dz'. '/api/executePgwpfe3dsPayment',
            json_decode($payment->request_data, true));
        return response()->json(
            [
                'status' => true,
                'return_url' => 'http://'.$payment->return_url,
                'request_data' => $payment->request_data,
                'message' => $message->message,
                'dd' => json_decode((string)$response->getBody())
            ]
            , 200);
    }
    return response()->json([
        'status' => false,
        'message' => $message->message
    ], 200);
}
public function ReceiveCReq(Request $request)
{
    return response()->json([
        'message' => '3DS protocol initiated successfully'
    ], 200);
}

```

## B.2. MPI (Merchant Plug-In)

### a. 3D Secure initiation

```

// added by nasreddine & lahcen
// this code is used to send 3D Secure initiation from
// MPI(Merchant Plug-In) to Payment gateway
if ($request->input('payment') == "pgw-pfe3ds"){
    $params['form_params'] = [
        'intent' => "sale",
        'merchant' => "hanouti",
        'merchant_website' => "http://hanouti.dz",
        'return_url' =>
        "www.hanouti.dz/returnpgwpfe3ds?payment=pgw-pfe3ds&billing_address=12",
        'cancel_url' => "www.hanouti.dz/checkout/cancel",
        'payment_method' => "pgw-pfe3ds",
        'discounts' => 0,
        'customer_id' => Auth::user()->id,

```



```

        'total_products' => $cartRepo->getSubTotal(),
        'total' => $cartRepo->getTotal(),
        'total_paid' => $cartRepo->getTotal(2, $shippingFee),
        'tax' => $cartRepo->getTax(),
        'billing_address' => $request->input('billing_address'),
        'transactions' => json_decode($this->paypal->getTransactions())
    ];
    $url = config('pgw-pfe3ds.api_url')."/api/init";
    $client = new Client(['base_uri' => config('pgw-pfe3ds.api_url')]);
    $response = $client->post($url, $params);
    $redirectUrl = config('pgw-pfe3ds.api_url');
    return redirect()->to('http://'.$redirectUrl);
}

```

## b. Return methods from payment gateway to online shop

```

/**
 * Cancel page
 *
 * @param Request $request
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function cancel(Request $request)
{
    $cartRepo = new CartRepository(new ShoppingCart);
    $cartRepo->clearCart();
    return view('front.checkout-cancel', ['data' => $request->all()]);
}
/**
 * Success page
 *
 * @return \Illuminate\Contracts\View\Factory|\Illuminate\View\View
 */
public function success()
{
    $products = $this->cartRepo->getCartItems();
    $customer = Customer::find(15); // $request->user();
    $rates = null;
    $shipment_object_id = null;
    if (env('ACTIVATE_SHIPPING') == 1) {
        $shipment = $this->createShippingProcess($customer, $products);
        if (!is_null($shipment)) {
            $shipment_object_id = $shipment->object_id;
            $rates = $shipment->rates;
        }
    }
    // Get payment gateways
    $paymentGateways = collect(explode(',', config('payees.name')))
        ->transform(function ($name) {return config($name);})->all();
    $billingAddress = $customer->addresses()->first();
    $invoice_code = mt_rand(1000, 9999);
    $results = [
        'customer' => $customer,
        'date' => now(),
        'invoice_code' => $invoice_code,
        'billingAddress' => $customer->addresses()->first(),
        'addresses' => $customer->addresses()->get(),
        'products' => $this->cartRepo->getCartItems(),
        'subtotal' => $this->cartRepo->getSubTotal(),
        'tax' => $this->cartRepo->getTax(),
        'total' => $this->cartRepo->getTotal(2),
        'payments' => $paymentGateways,
        'cartItems' => $this->cartRepo->getCartItemsTransformed(),
        'shipment_object_id' => $shipment_object_id,
        'rates' => $rates
    ];
}

```

```

];
// Send email to customer
$this->SendEmail($customer->full_name,
$customer->email,
'New purchases',
$results);
// Clear cart
$cartRepo = new CartRepository(new ShoppingCart);
$cartRepo->clearCart();
return view('front.checkout-success',$results);
}
// added by Lahcen Bouchrit & Nasr Eddine uelfout, 2020
// Execute this method when we returned from Payment gateway
public function executePgwPfe3dsPayment(Request $request)
{
    try {
        $order = new Order([
            'reference' => Uuid::uuid4()->toString(),
            'courier_id' => 1,
            'customer_id' => $request->input('customer_id'),
            'address_id' => $request->input('billing_address'),
            'order_status_id' => 1,
            'payment' => $request->input('payment_method'),
            'discounts' => $request->input('discounts'),
            'total_products' => $request->input('total_products'),
            'total' => $request->input('total'),
            'total_paid' => $request->input('total_paid'),
            'tax' => $request->input('tax'),
            'invoice' => '',
            'label_url' => '',
            'tracking_number' => '',
            'total_shipping' => 0
        ]);
        $order->save();
        $transactions = $request->transactions;
        $items = $transactions['item_list']['items'];
        foreach($items as $item){
            $product = Product::where('name', '=', $item['name'])->first();
            $order_product = new OrderProduct([
                'order_id' => $order->id,
                'product_id' => $product->id,
                'product_attribute_id' => null,
                'quantity' => $item['quantity'],
                'product_name' => $product->name,
                'product_sku' => $product->sku,
                'product_description' => $product->description,
                'product_price' => $product->price
            ]);
            $order_product->save();
        }
        $this->cartRepo->clearCart();
        return response()->json([
            'statusCode' => '200',
            'message' => 'success'
        ], 200);
    } catch (\Exception $e) {
        return response()->json([
            'statusCode' => '200',
            'message' => 'error',
            'error' => $e->getMessage()
        ], 200);
    }
}
public function returnPgwPfe3ds(Request $request)

```

```

{
    return redirect()->route('checkout.success');
}

```

## B.3. Issuer requirements

### a. Library

```

<?php
namespace App\PGW;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Http;
use App\Mail\MailSender;
use Illuminate\Support\Facades\Mail;
trait PgwMethods
{
    protected function GetLastPaymentInfo()
    {
        return DB::table('payments')->latest()->first();
    }
    protected function GenerateCode()
    {
        return mt_rand(10000, 99999);
    }
    protected function SendEmailTo($to_name, $to_email, $code)
    {
        $data = array('name'=>$to_name,
            'body' => 'Verification mail, your code is: '.$code);
        Mail::send('emails.mail', $data,
            function($message) use ($to_name, $to_email) {
                $message->to($to_email, $to_name)->subject('Verefication code');
                $message->from('issuer_bank@gmail.com', 'issuer_bank');
            });
    }
    protected function SendEmailToCustomer($full_name, $to_email)
    {
        $data = array('name'=>$full_name,
            'body' => 'Thank you so much for allowing us to help you
with your recent account opening. We are committed to providing our customers with
the highest level of service and the most innovative banking products possible.

                We are very glad you chose us as your financial institution
and hope you will take advantage of our wide variety of savings, investment and
loan products, all designed to meet your specific needs.');
```

```

    $client = new Client(['base_uri' => config('pgw-
pfe3ds.api_url')."/api/pgw-pfe3ds"]);
    $params['form_params'] = [
        "intent" => "sale",
        "merchant" => "hanouti",
        "merchant_website" => "http://hanouti.dz",
        "return_url" =>
"www.hanouti.dz/checkout/execute?payment=pgw-pfe3ds&billing_address=12",
        "cancel_url" => "www.hanouti.dz/checkout/cancel",
        "payment_method" => "pay-pfe-3ds",
        'discounts' => 0,
        'total_products' => $cartRepo->getSubTotal(),
        'total' => $cartRepo->getTotal(),
        'total_paid' => $cartRepo->getTotal(2, $shippingFee),
        'tax' => $cartRepo->getTax()
    ];
    $url = config('pgw-pfe3ds.api_url')."/api/pgw-pfe3ds/init";
    $response = $client->post($url,$params);
    $body = $response->getBody();
    $html_string = (string) $body ;
}
}

```

## b. API Methods

```

<?php
namespace App\Http\Controllers;
use App\Payment;
use App\Transaction;
use App\Card;
use Illuminate\Http\Request;
use App\PGW\PgwMethods;

class ApiController extends Controller
{
    //
    use PgwMethods;
    private $email;
    private $code;
    //
    public function ReceiveAReq(Request $request){
        $messageAReq = $request->all();
        $interoperabilityURL = $request->dsURL;
        $card = Card::where('number', '=', $request->acctNumber)->first();
        if (!empty($card)){
            if ($request->cardholderName != $card->cardholder_name){
                $messageARes = [
                    'statusCode' => '401',
                    'field' => 'cardholderName',
                    'message' => 'Incorrect Cardholder name'
                ];
                return response()->json($messageARes, 401);
            }
            if ($request->cardExpiryDate != $card->expiry_date){
                $messageARes = [
                    'statusCode' => '401',
                    'field' => 'cardExpiryDate',
                    'message' => 'Incorrect expiry date'
                ];
                return response()->json($messageARes, 401);
            }
            if ($request->purchaseAmount >
                $card->account()->first()->current_balance){
                $messageARes = [
                    'statusCode' => '401',

```

```
        'field' => 'cvCode',
        'message' =>
'Insufficient account balance, balance less then total amount paid'
    ];
    return response()->json($messageARes, 401);
}
if ($request->cvCode != $card->cvv){
    $messageARes = [
        'statusCode' => '401',
        'field' => 'cvCode',
        'message' => 'Incorrect cvv2/cvc2 code'
    ];
    return response()->json($messageARes, 401);
}
// Sending ARes
$messageARes = [
    'statusCode' => '200',
    'message' => 'success',
    'acsURL' => "www.issuer.bank.dz",
    'authenticationType' => "email",
    'acctID' => $card->account()->first()->id,
    'email' => $card->account()->first()->customer()->first()->email,
    'mobilePhone' => $card->account()->first()->customer()->first()->phone
];
$this->email = $messageARes["email"];
return response()->json($messageARes, 200);
}
else {
    $messageARes = [
        'statusCode' => '401',
        'field' => 'cardNumber',
        'message' => 'Incorrect card number, Card dosn\'t exist'
    ];
    return response()->json($messageARes, 401);
}
// Acs --> Inter
return response()->json([
    'statusCode' => '401',
    'message' => 'Failer from issuer bank'
], 401);
}
public function ReceiveCReq(Request $request){
    $messageCReq = $request->all();
    // Sending CRes
    $messageCRes = [
        "threeDSServerTransID" => "",
        "acsCounterAtoS" => "",
        "acsTransID" => "",
        "acsHTML" => "<test>",
        "acsUiType" => "",
        "challengeCompletionInd" => "",
        "challengeInfoHeader" => "",
        "challengeInfoLabel" => "",
        "challengeInfoText" => "",
        "challengeInfoTextIndicator" => "",
        "challengeSelectInfo" => "",
        "expandInfoLabel" => "",
        "expandInfoText" => "",
        "issuerImage" => "",
        "messageExtension" => "",
        "messageType" => "",
        "messageVersion" => "",
        "oobAppURL" => "",
```

```

        "oobAppLabel" => "",
        "oobContinueLabel" => "",
        "psImage" => "",
        "resendInformationLabel" => "",
        "sdkTransID" => "",
        "submitAuthenticationLabe" => "",
        "whitelistingInfoText" => "",
        "whyInfoLabel" => "",
        "whyInfoText" => "",
    ];
    return response()->json($messageCRes, 200);
}
public function SendEmail(Request $request){
    $card = Card::where('number', '=', $request->cardNumber)->first();
    if (!empty($card)){
        $card->code = $this->GenerateCode();
        $card->update();
        $this->SendEmailTo("customer", $request->email, $card->code);
        return response()->json([
            'message' => 'Email sent successfully'
        ], 200);
    }
    return response()->json([
        'message' => 'error'
    ], 200);
}
public function VerifyCode(Request $request){
    $card = Card::where('number', '=', $request->cardNumber)->first();
    if ($card->code == $request->code){
        //sending RReq
        $messageRReq = [
            "threeDSServerTransID" => "",
            "acsTransID" => "",
            "acsRenderingType" => "",
            "authenticationMethod" => "",
            "authenticationType" => "",
            "authenticationValue" => "",
            "challengeCancel" => "",
            "dsTransID" => "",
            "eci" => "",
            "interactionCounter" => "",
            "messageCategory" => "",
            "messageExtension" => "",
            "messageType" => "",
            "messageVersion" => "",
            "sdkTransID" => "",
            "transStatus" => "",
            "transStatusReason" => "",
            "whiteListStatus" => "",
            "whiteListStatusSource" => "",
            'statusCode' => '200',
            'merchant' => $request->merchant,
            'amount' => $request->amount,
            'message' => 'success',
            "acsURL" => "www.issuer.bank.dz",
            "authenticationType" => "email",
            "acctID" => $card->account()->first()->id,
            "email" => $card->account()->first()->customer()->first()->email,
            "mobilePhone" => $card->account()->first()->customer()->first()->phone
        ];
        $url =
"www.interbank.network.dz"."/api/interoperability/receiverreq";
        $response = $this->SendRReq($url, $messageRReq);
    }
}

```

```

$messageRRes = json_decode((string)$response->getBody());
//create debit transaction
$account = $card->account()->first();
$transaction = new Transaction([
    'account_id' => $account->id,
    'debit' => $request->amount,
    'credit' => 0,
    'current_balance' => $account->current_balance,
    'type' => 'Online',
    'currency_code' => 'DZD',
    'description' => ''
]);
$transaction->save();
$account->current_balance = $account->current_balance -
$transaction->debit + $transaction->credit;
$account->update();
$card->code = '';
$card->update();
return response()->json([
    'statusCode' => '200',
    'message' => 'The code is correcte'
], 200);
}
return response()->json([
    'statusCode' => '401',
    'message' => 'The code '.$request->code.' is incorrecte'
], 401);
}
}

```

## B.4. Acquirer requirements

### a. Library

```

<?php
namespace App\PGW;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Http;
use App\Mail\MailSender;
use Illuminate\Support\Facades\Mail;
trait PgwMethods
{
    protected function GetLastPaymentInfo()
    {
        return DB::table('payments')->latest()->first();
    }
    protected function SendAReq($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
    protected function SendEmailToCustomer($full_name, $to_email)
    {
        $data = array('name'=>$full_name,
            'body' => 'Congratulation! Mr. '.$full_name.', You have
create a new bank account');
        Mail::send('emails.mail', $data,
function($message) use ($full_name, $to_email) {
    $message->to($to_email, $full_name)->subject('New Account');
    $message->from('acquirerbank@gmail.com', 'acquirer_bank');
});
    }
}
}

```

## b. API Methods

```

<?php
namespace App\Http\Controllers;
use App\Payment;
use App\Transaction;
use App\Customer;
use Illuminate\Http\Request;
use App\PGW\PgwMethods;
class ApiController extends Controller
{
    //
    use PgwMethods;
    //
    public function ReceiveAReq(Request $request) {
        $message = $request->all();
        $dsURL = $request->dsURL;
        // Acq --> Inter
        $response = $this->SendAReq($dsURL, $message);
        return json_encode(json_decode((string)$response->getBody()));
    }
    public function ReceiveRReq(Request $request) {
        try{
            //create debit transaction
            $customer = Customer::where('full_name', '=',
                $request->merchant)->first();
            if (empty($customer)){
                $customer = Customer::where('first_name', '=',
                    $request->merchant)->first();
            }
            $account = $customer->account()->first();
            $transaction = new Transaction([
                'account_id' => $account->id,
                'debit' => 0,
                'credit' => $request->amount,
                'current_balance' => $account->current_balance,
                'type' => 'Online',
                'currency_code' => 'DZD',
                'description' => ''
            ]);
            $transaction->save();
            $account->current_balance = $account->current_balance -
            $transaction->debit + $transaction->credit;
            $account->update();
            // send RRes
            $messageRRes = [
                "threeDSServerTransID" => "",
                "acsTransID" => "",
                "acsRenderingType" => "",
                "authenticationMethod" => "",
                "authenticationType" => "",
                "authenticationValue" => "",
                "challengeCancel" => "",
                "dsTransID" => "",
                "eci" => "",
                "interactionCounter" => "",
                "messageCategory" => "",
                "messageExtension" => "",
                "messageType" => "",
                "messageVersion" => "",
                "sdkTransID" => "",
                "transStatus" => "",
                "transStatusReason" => "",
                "whiteListStatus" => "",
            ]
        }
    }
}

```



```

        "whiteListStatusSource" => "",
        'statusCode' => '200',
        'message' => 'success',
        "acsURL" => "www.issuer.bank.dz",
        "authenticationType" => "email"
    ];
    return response()->json($messageRRes, 200);
} catch (\Exception $e) {
    return response()->json(['message' => $e->getMessage()], 200);
}
}
}

```

## B.5. Interoperability requirements

### a. API Methods

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\PGW\PgwMethods;

class ApiController extends Controller
{
    use PgwMethods;
    public function ReceiveAReq(Request $request) {
        $message = $request->all();
        $issuerURL='www.issuer.bank.dz/api/issuer/receiveareq';
        $response = $this->SendAReq($issuerURL, $message);
        return json_encode(json_decode((string)$response->getBody()));
    }

    public function ReceiveRReq(Request $request){
        $message = $request->all();
        $acquirerURL='www.acquirer.bank.dz/api/acquirer/receiverreq';
        $response = $this->SendRReq($acquirerURL, $message);
        return json_encode(json_decode((string)$response->getBody()));
    }
}

```

### b. Controller

```

<?php
namespace App\PGW;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Http;

trait PgwMethods
{
    protected function SendAReq($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
    protected function SendRReq($url, $message)
    {
        $response = Http::post($url, $message);
        return $response;
    }
}
}

```