ENSICAEN
6, bd maréchal Juin
F-14050 Caen cedex 4

Spécialité Informatique - 2e année

Rapport de projet

Étude de la sécurité de 3D secure

Samuel AUGUSTE
Baptiste GIROUD

Suivi Ensicaen : Patrick LACHARME
Sylvain VERNOIS

2010 - 2011

Table des matières

Introduction

La problématique liée au commerce électronique est importante, car elle comporte des enjeux importants : le marchand doit être payé, le client veut s'assurer que le service sera fourni par le marchand, et enfin, la banque doit s'assurer qu'il n'y a pas d'usurpation d'identité afin de protéger son client. Les données mises en jeu doivent donc rester confidentielles (numéro de carte bancaire …).

Un tel échange est complexe, car il implique plusieurs acteurs (client, marchand, banques), et de nombreux aspects techniques ou scientifiques (sécurité informatique, base de données, cartes à puces …).

Dans un premier temps, nous allons voir quels sont les moyens qui peuvent être mis en œuvre pour sécuriser un échange d'informations, puis les différents protocoles proposés pour tenter de sécuriser un achat en ligne, en particulier celui actuellement en place, le protocole 3D secure. [1-2]

Ensuite, nous allons nous attacher à expliquer le travail réalisé au second semestre. Celui-ci visait à développer une application pour simuler un échange 3D secure en créant les différents messages mis en jeu.

I) Étude du protocole 3D secure.

Dans un premier temps, nous avons cherché à mieux connaître le protocole 3D secure, son rôle et ses origines. Cette étape de recherche et d'analyse documentaire a été l'objectif du premier semestre, et nous a permis de comprendre le protocole, que nous détaillons dans cette première partie.

1 Solutions pour sécuriser un échange

La base d'un échange sécurisé repose sur l'authentification des différents protagonistes.

Pour cela, le client possède déjà quelques éléments (carte bancaire, informations personnelles …). Le cryptogramme visuel (3 derniers numéros au dos de la carte bancaire) est un exemple de moyen supplémentaire mis en œuvre pour l'authentification, bien qu'il soit insuffisant (en cas de vol de la carte).

Quel que soit l'échange, différentes méthodes peuvent être mises en œuvre pour améliorer la sécurité en permettant l'authentification. Voyons tout d'abord la méthode couramment employée pour la gestion de certificats : l'infrastructure à clés publiques.

1.1 Infrastructure à clés publiques :

Une infrastructure à clés publiques (ICP), est un ensemble de composants physiques (ordinateurs, cartes à puces, algorithmes et équipements cryptographiques), de procédures humaines (vérifications, validation) et de logiciels pour gérer le cycle de vie de certificats électroniques.

Le chiffrement s'effectuant par l'utilisation de clés asymétriques, un certificat électronique permet de garantir le lien entre une clé publique et son possesseur, en vue de pouvoir l'authentifier facilement.

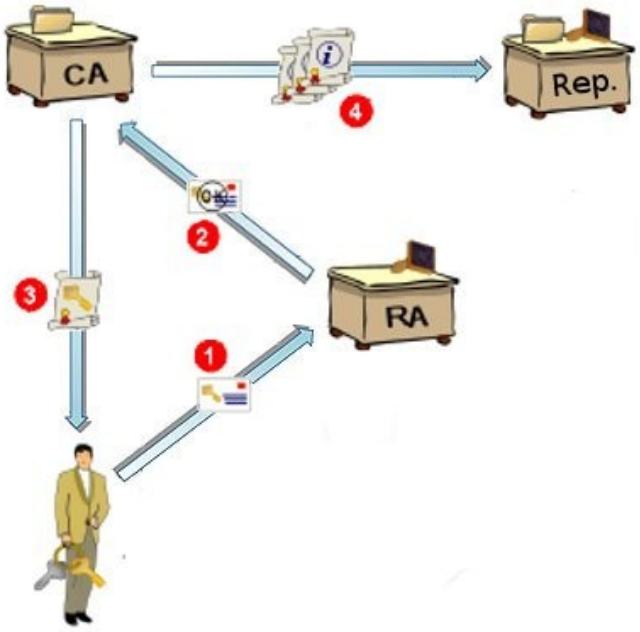


Fig 1 – Infrastructure à clés publiques [3]

1) Tout d'abord, l'utilisateur final fait une demande de certificat électronique, auprès d'une Autorité d'Enregistrement (RA), et lui fournit des informations justifiant son identité.
2) L'autorité d'enregistrement vérifie cette identité, puis génère une paire de clés. La clé privée est destinée à l'utilisateur, et la clé publique est remise à l'Autorité de Certification (CA).
3) L'autorité de certification signe ensuite le certificat, à l'aide de sa propre clé privée. Celui-ci peut ainsi être utilisé pour authentifier l'utilisateur.
4) Enfin, L'autorité de certification (CA) envoie le certificat à l'Autorité de dépôt (*repository*), qui stocke également les listes de révocation.

L'usage de certificats générées par une infrastructure à clés publiques est donc intéressant pour les transactions électroniques. En permettant l'authentification, elle peut servir à garantir une certaine sécurité dans un échange d'informations (les données ne sont pas envoyées par n'importe qui ou à n'importe qui).

Nous allons à présent voir un protocole qui utilise une infrastructure à clés publiques, dans le but de sécuriser un échange.

1.2 Le protocole SSL

Le SSL (Secure Sockets Layer), est un protocole pour sécuriser l'échange d'informations, utilisé par HTTPS (Hyper Text Transfert Protocol Secured). En particulier, dans le cas du commerce en ligne, il permet de sécuriser dans une certaine mesure l'échange de données entre clients et marchands. Il peut se décomposer en quatre principales étapes :

1) Grâce à l'utilisation d'une infrastructure à clés publiques, le marchand possède sa clé privée, ainsi qu'une clé publique, qu'il transmet au client.
2) Pour sécuriser l'envoi de ses données, le client va utiliser un algorithme générant aléatoirement une nouvelle clé. Il envoie celle-ci au marchand en la chiffrant, grâce à la clé publique de ce dernier, qu'il a précédemment récupérée. La suite de la transaction sera chiffrée avec cette clé secrète, il s'agit donc de chiffrement symétrique.
3) Pour vérifier que l'échange précédent s'est bien déroulé, le client va fournir au marchand une information quelconque, qu'il chiffre avec la clé secrète, précédemment créée. Le marchand s'authentifie auprès du client en déchiffrant l'information, et en la lui renvoyant en clair.
4) Enfin, le client envoie au marchand le numéro, la date d'expiration et le cryptogramme visuel de sa carte bancaire, en les chiffrant à l'aide de la clé secrète, qu'ils sont les seuls à posséder. Le marchand peut alors lui même utiliser ces informations pour débiter le compte du client et finaliser la transaction.

L'usage de SSL permet donc :

- La confidentialité, concernant les informations envoyées, puisqu'elles sont chiffrées.
- L'authentification du marchand au début de la transaction.
- L'intégrité des données échangées.

Cependant, l'usage de ce système est limité :

- Les données sont vulnérables face à une attaque de type homme du milieu, puisqu'il n'y a pas d'authentifications au cours de la transaction.
- Il ne permet pas la non-répudiation par le client, la clé étant connue par plusieurs personnes.
- Le marchand reçoit au final toutes les coordonnées bancaires du client en clair, ce qui pose problème si ce dernier est malintentionné.

Voyons maintenant les protocoles qui ont étés développés spécifiquement pour le commerce électronique, par les entreprises de paiement, et qui sont plus rigoureux en matière d'authentification.

2 Protocoles pour le commerce électronique

2.1 Le protocole SET

Pour faire face à ces manquements en matière de sécurité, Visa et MasterCard ont développés le protocole SET (Secure Electronic Transactions), ancêtre du 3D-secure.

SET fait intervenir de nouveaux acteurs : une passerelle de paiement, faisant office d'intermédiaire entre internet et le réseau de cartes de paiement, ainsi qu'une autorité de certification maitresse, qui a pour rôle d'assurer l'authentification de tous les intervenants. [4]

Le protocole peut être décrit comme 4 échanges principaux, tous transmis en SSL :

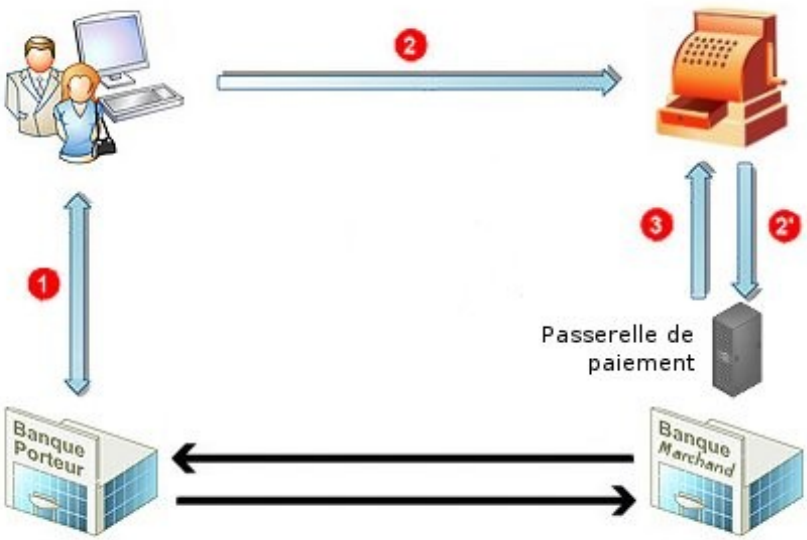


Fig 2 – Protocole SET

1) Enregistrement du client. Le client fait une demande d'enregistrement auprès de sa banque, qui lui sert ainsi d'autorité de certification. La banque se charge de vérifier son identité. Ensuite, le client crée sa paire de clé, et communique sa clé publique à sa banque. Celle-ci créée, signe et lui fait parvenir son certificat.
2) L'ordre d'achat. Le marchand possède lui aussi un certificat. Le client l'authentifie et lui fait parvenir sa demande d'achat, ainsi que ses informations de paiement. Le marchand se charge de retransmettre le message contenant les informations de paiement, après en avoir vérifié l'intégrité, à la passerelle de paiement.
3) Paiement. La passerelle de paiement vérifie les données bancaires, puis autorise le marchand à finaliser la transaction. Ce dernier répond donc à la demande d'achat du client, et peut ensuite demander le paiement à la passerelle.

Ce protocole garantit l'intégrité des données échangées à tous niveaux, ainsi que l'authentification du client, du marchand et de la passerelle de paiement.

Un des avantages réside dans le renforcement de la confidentialité. Le client, le marchand comme la passerelle de paiement possèdent chacun un certificat. Cela permet de mettre en pratique la méthode dite de signature duale, c'est à dire que le client envoie un message contenant sa demande d'achat et ses informations bancaires, mais que la passerelle de paiement n'a pas accès à l'ordre d'achat du client. De même, le marchand ne pourra pas accéder aux coordonnées bancaires du client, ce qui résout un problème précédemment évoqué.

De plus, ce protocole répond à la problématique de non-répudiation. Grâce à son certificat, le client est authentifié lorsqu'il passe son ordre d'achat, et il ne peut donc pas le contester par la suite. De même, le client enregistre la réponse du marchand concernant sa demande d'achat.

Cependant, ce système est pénalisant pour le marchand. En effet, c'est lui qui se charge de transmettre les coordonnées bancaires du client à la passerelle de paiement. Bien que pouvant vérifier l'intégrité de ce message, il ne peut pas en vérifier les données. Il est donc désigné comme responsable en cas d'informations frauduleuses fournies par le client.

Pour cette raison, certaines sociétés (PayPal, Digicash …) se proposent comme intermédiaires entre le client et le marchand. Ce sont donc elles qui payent le marchand et qui prélèvent le client, s'octroyant au passage une commission. La contrainte est grande, car un marchand et un client doivent pour cela tous deux être inscrits auprès du même intermédiaire et lui faire pleinement confiance, mais le marchand a l'assurance d'être payé.

Dans le cadre du protocole SET, le client doit posséder de son coté un logiciel pour gérer sa demande de certificat et ses clés. De plus, l'utilisation de ce protocole nécessite une infrastructure de certification poussée. En pratique, cela est complexe et très lourd à mettre en place, ce qui fait que SET n'a été que très peu utilisé. Visa et MasterCard lui ont très vite cherché un successeur.

2.2 Fonctionnement du protocole 3D secure

La solution actuellement mise en œuvre pour le paiement en ligne est le protocole 3D secure. Celui ci a été lancé par Visa en 2001, mais est également utilisé par MasterCard. Le terme 3D provient des 3 domaines où s'effectuent les authentifications, qui sont :

- Le marchand et sa banque, qui recevra les fonds (*Acquirer Domain*).
- Le client et sa banque, qui a délivré la carte de paiement (*Issuer Domain*).
- Le système de carte bancaire (*Interoperability Domain*).

3D secure effectue donc toutes les authentifications, mais n'utilise pas de hiérarchie avec autorité de certification maîtresse, ni de passerelle de paiement. La nouveauté est l'intervention du serveur Visa. Il permet à Visa de réaliser les fonctions suivantes :

- Un service d'annuaire, pour déterminer la banque d'un client et son numéro de compte à partir de sa carte.
- Une autorité de certification, qui génère les différents certificats.
- Un historique de toutes les tentatives d'authentifications réalisées au cours de transactions passées.

Un échange 3D secure est constitué de messages XML, acheminés en SSL [5] :

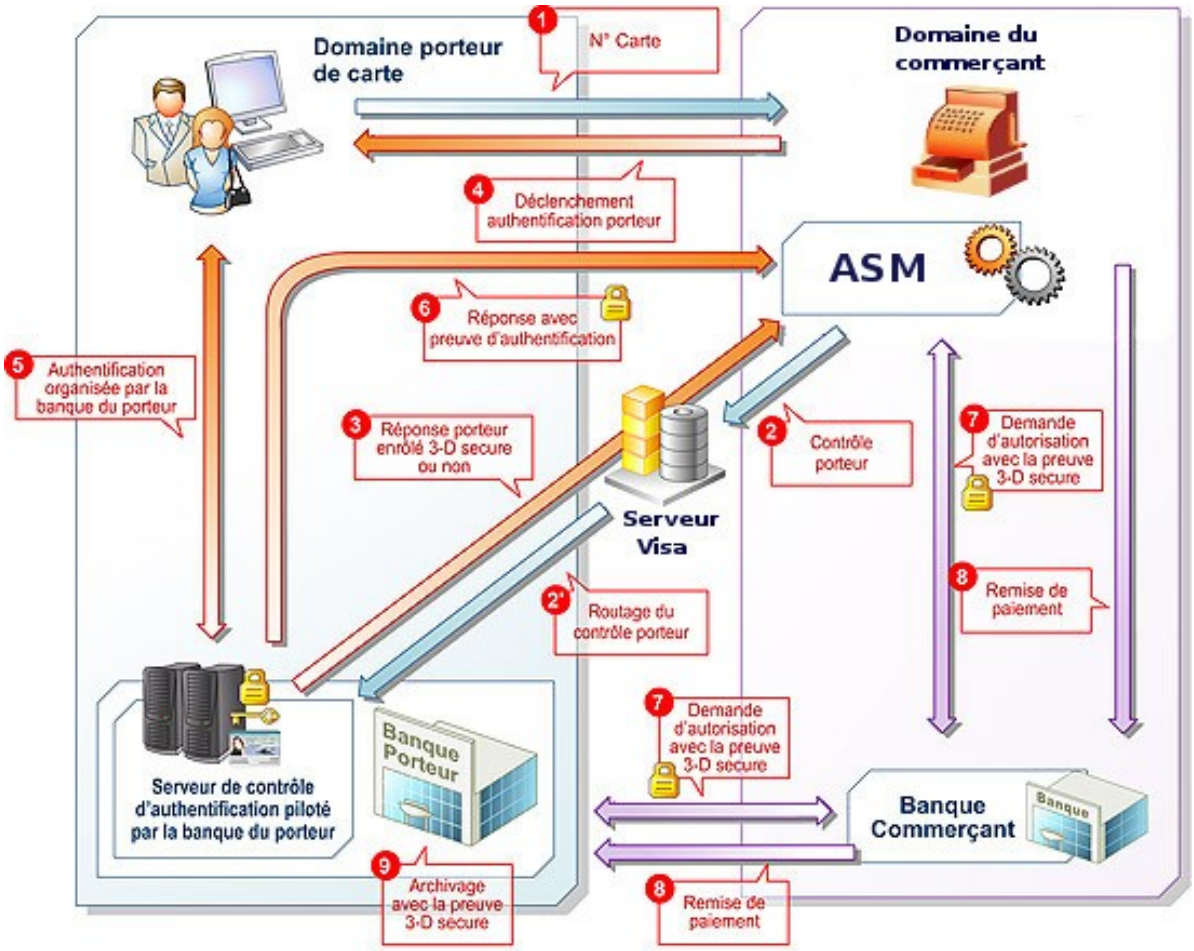


Fig 3 – Protocole 3D secure [6]

1) Le client envoie au marchand son intention d'achat, et lui fournit ses coordonnées de carte bancaire (numéro, date d'expiration et cryptogramme visuel).
2) Pour communiquer avec le serveur Visa, le marchand doit disposer d'un ajout au serveur marchand (ASM), plug-in qu'il se procure auprès de Visa ou MasterCard.
   L'ASM interroge l'annuaire Visa à l'aide du message VEReq (Verify Enrollment Request). Le serveur Visa se charge alors d'authentifier le marchand, le numéro de carte et la banque du client.

3) La réponse au message précèdent est contenue dans le message VERes (Verify Enrollment Result). Le serveur de contrôle d'authentification (SCA) de la banque y indique si le client est inscrit au programme 3D secure, et transmet au marchand son URL.
4) Suite à cette réponse, l'ASM du marchand émet le message PAReq (Payer Authentification Request) vers l'URL du SCA obtenue à l'étape précédente. Ce message contient les détails de l'achat à autoriser.
5) L'ASM transfère ensuite le contrôle à la banque, en ouvrant chez le client une fenêtre pop-up vers le SCA de la banque, obtenue dans l'URL précédente. La banque se charge alors de l'authentification de son client, en lui demandant les informations qu'elle jugera nécessaire.
6) Pour confirmer l'authentification du client, le SCA envoie le message PARes à l'ASM.
7) Le marchand est donc en mesure de demander une autorisation de paiement à sa banque. Après authentification du marchand par sa banque, et vérification de la nature de la transaction auprès de la banque du client, elle confirme l'autorisation de paiement auprès du marchand.
8) Le marchand obtient donc le paiement de la transaction.
9) Les informations qui ont permis le paiement sont stockées par la banque du client, afin d'assurer la non-répudiation de la transaction par les différentes parties.

Ce système présente quelques contraintes comme la nécessité pour le marchand de payer pour l'utilisation d'un ASM, ou celle de la banque de posséder un SCA.

Néanmoins, le protocole 3D secure présente l'avantage important de faire reposer la totalité de la sécurité de la transaction sur la banque, et non sur le marchand, qui a alors la garantie de paiement.

2.3 Utilisation du 3D secure

La responsabilité de la transaction reposant sur la banque, c'est à elle d'authentifier le client. Pour cela, elle est libre de choisir les méthodes de sécurité qu'elle juge suffisantes. Voici quelques exemples d'informations demandées en pratique [7] :

- Date de naissance (Caisse d'Épargne).
- Code secret à usage unique, possédé par le client (CIC et Crédit Mutuel).
- Code secret à usage unique, reçu par SMS (Axa, HSBC, BNP Paribas, Société Générale, Groupama, Crédit Agricole dont LCL, …).

La demande d'un code à usage unique, reçu par SMS, est une évolution relativement récente pour la plupart des banques. En effet, la Banque de France a envoyé une recommandation durant l'été 2010, demandant aux banques de garantir une authentification forte dans les transactions de paiement en ligne. Le téléphone portable étant un objet courant, possédé uniquement par le client, et devant le faible coût d'un SMS, c'est la méthode qui a été retenue par la plupart des banques. Cette uniformisation de la méthode d'authentification a souvent remplacé l'usage de la date de naissance, qui n'est pas une information confidentielle, surtout à l'heure des réseaux sociaux.

Les sites utilisant le protocole 3D secure sont identifiables grâce aux logos suivants :

Malgré les avantages qu'offre au marchand le protocole 3D secure, nous pouvons constater que ce protocole est peu utilisé chez les marchands français. Cela s'explique principalement par l'arrivée brutale de ce protocole en France, que les banques ont effectuée sans en informer suffisamment les clients. Ainsi, ces derniers ont pu être surpris et freinés par la demande de renseignements confidentiels dans une fenêtre pop-up. De plus, le client n'est pas toujours reconnu par le serveur Visa, ce qui conduit à l'échec de la transaction, le protocole 3D secure augmente la durée de la transaction, et il ne permet pas la paiement en plusieurs fois.

En pratique, le lancement du protocole 3D secure s'est donc traduit par une baisse du chiffre d'affaire pour les marchands. Même si cela affecte moins les petits marchands, on peut constater que sur les 15 sites de e-commerces les plus visités en France [8], aucun n'utilise le protocole 3D secure pour le paiement. La plupart du temps, c'est une simple connexion SSL qui est utilisée.

De plus, dans le cadre du protocole 3D secure, les clients se doivent d'être particulièrement vigilants, car ce sont leurs navigateurs qui permettent les échanges entre le marchand et leur banque (étapes 4, 5 et 6). Le client doit être en mesure de détecter toute anomalie, comme par exemple une tentative de phishing avec une fenêtre pop-up imitant sa banque. Pour cela, l'information du client est primordiale.

II) Réalisation d'une simulation.

Nous avons donc étudié d'un point de vue théorique le protocole 3D-Secure et son utilisation lors d'une transaction. Ces connaissances acquises, nous avons souhaité les mettre en application.

1 Étude d'une transaction.

L'ENSICAEN disposant d'une architecture 3D secure et un TP de 3ème année l'utilisant étant proposé, nous avons choisi de réaliser celui-ci, afin d'observer de manière effective le déroulement d'une transaction 3D secure, et l'échange de messages associés.

L'interface se décompose en deux parties. D'un coté nous avons le site du marchand que va voir le client, pour y passer une commande. De l'autre coté nous avons le BackOffice du marchand, où celui-ci peut récupérer et afficher toutes les informations sur le déroulement des transactions effectuées.

Pour acheter, le client entre son numéro de carte, sa date d'expiration et son cryptogramme visuel. Ensuite, un mot de passe lui est demandé, ce qui représente l'étape où celui-ci est authentifié par sa banque.

De son coté le marchand peut afficher toutes les transactions réalisées :

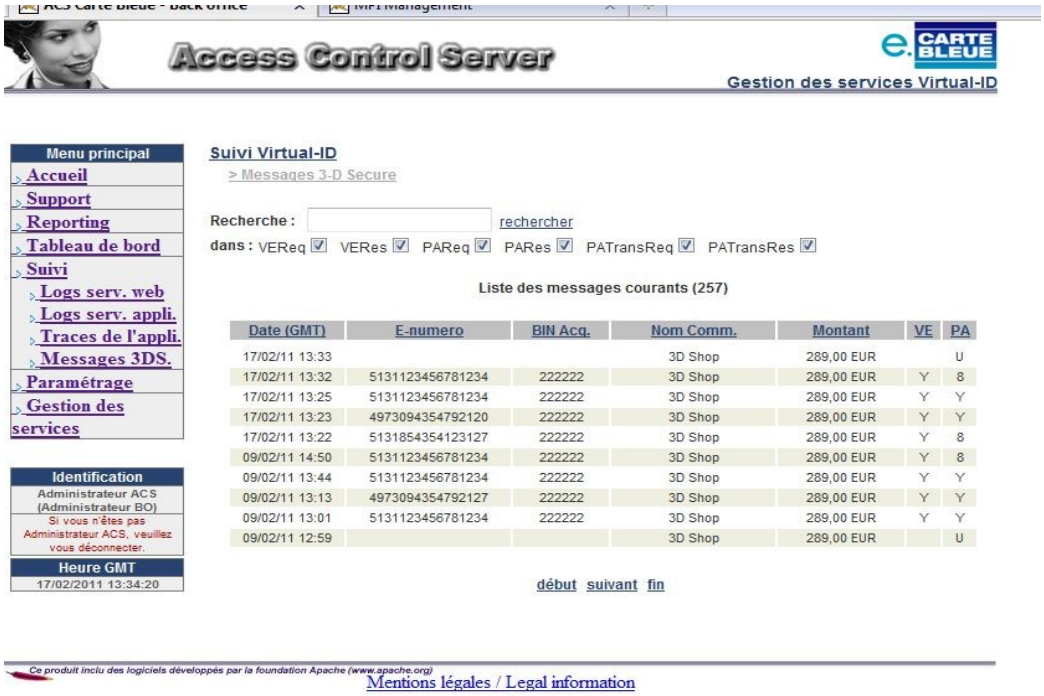


Fig 4 – Vue coté marchand.

Pour chacune, il peut choisir d'afficher indépendamment le détail de chaque message et le contenu du fichier XML associé (exemple avec PaReq :)

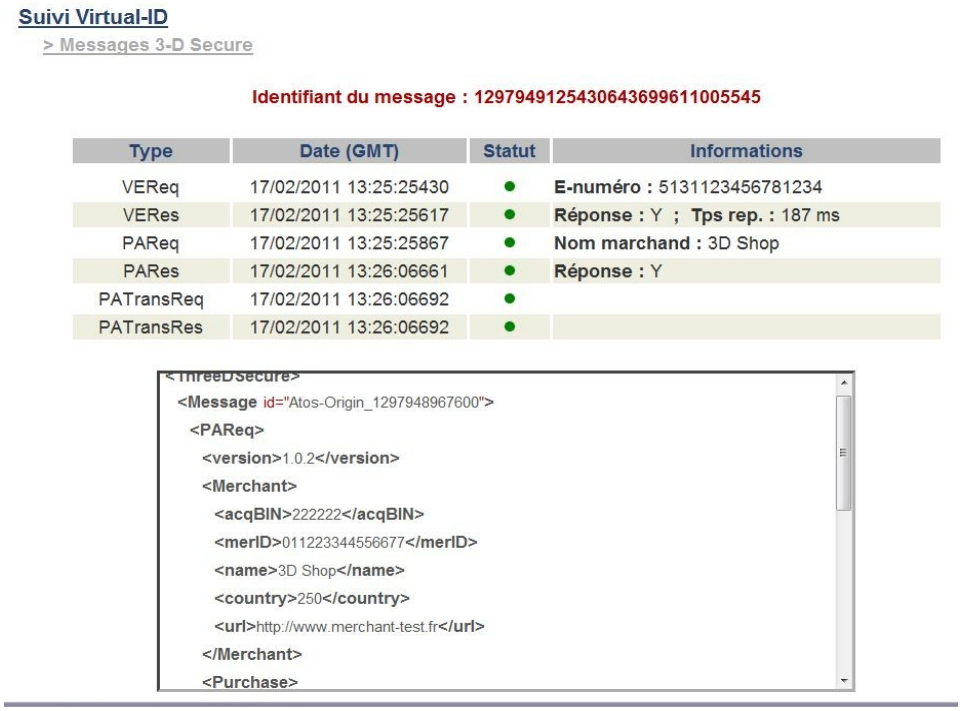


Fig 5 – Aperçu du message PaReq d'une transaction.

Nous avons effectué le TP en moins de 4 heures, ce qui nous a laissé le temps d'observer toutes les fonctionnalités offertes par cette maquette simulant la transaction 3D secure développée par Atos. Celle-ci est relativement limitée, puisque le mode d'authentification du client par la banque ce fait systématiquement par mot de passe, qui n'est jamais vérifié, toute saisie permettant d'effectuer l'achat. De plus, l'exemple du marchand n'est pas poussé, il n'est en effet possible de n'acheter qu'un seul article, à prix fixe, et aucune des données entrées par l'utilisateur (numéro de carte, cryptogramme) ne subit de vérifications.

L'application proposée est codée en Java, et d'après nos encadrants, la maquette étant prévue pour des démonstrations, il est difficile d'y effectuer des modifications simplement.

2 Proposition de modèle.

La deuxième partie du projet consistait donc à implémenter une architecture 3D secure en C#, de manière à ce que celle-ci présente un code propre, commenté et évolutif, afin que l'ajout de nouvelles fonctionnalités puisse se faire facilement.

Durant la phase d'analyse, nous avons donc réalisé des diagrammes pour préparer le développement de l'application :

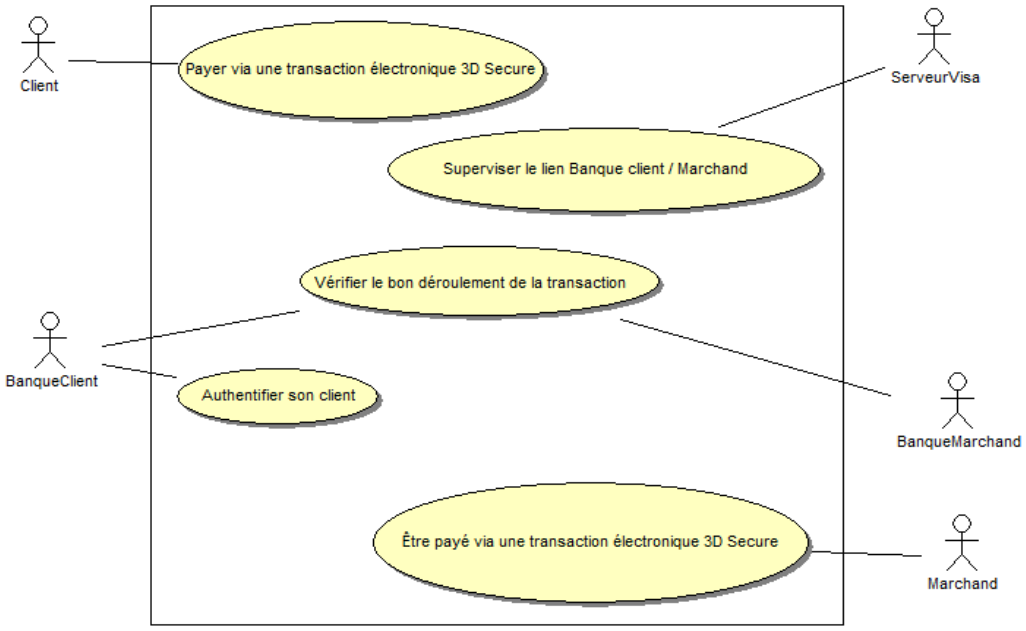


Fig 6 – Diagramme des cas d'utilisations.

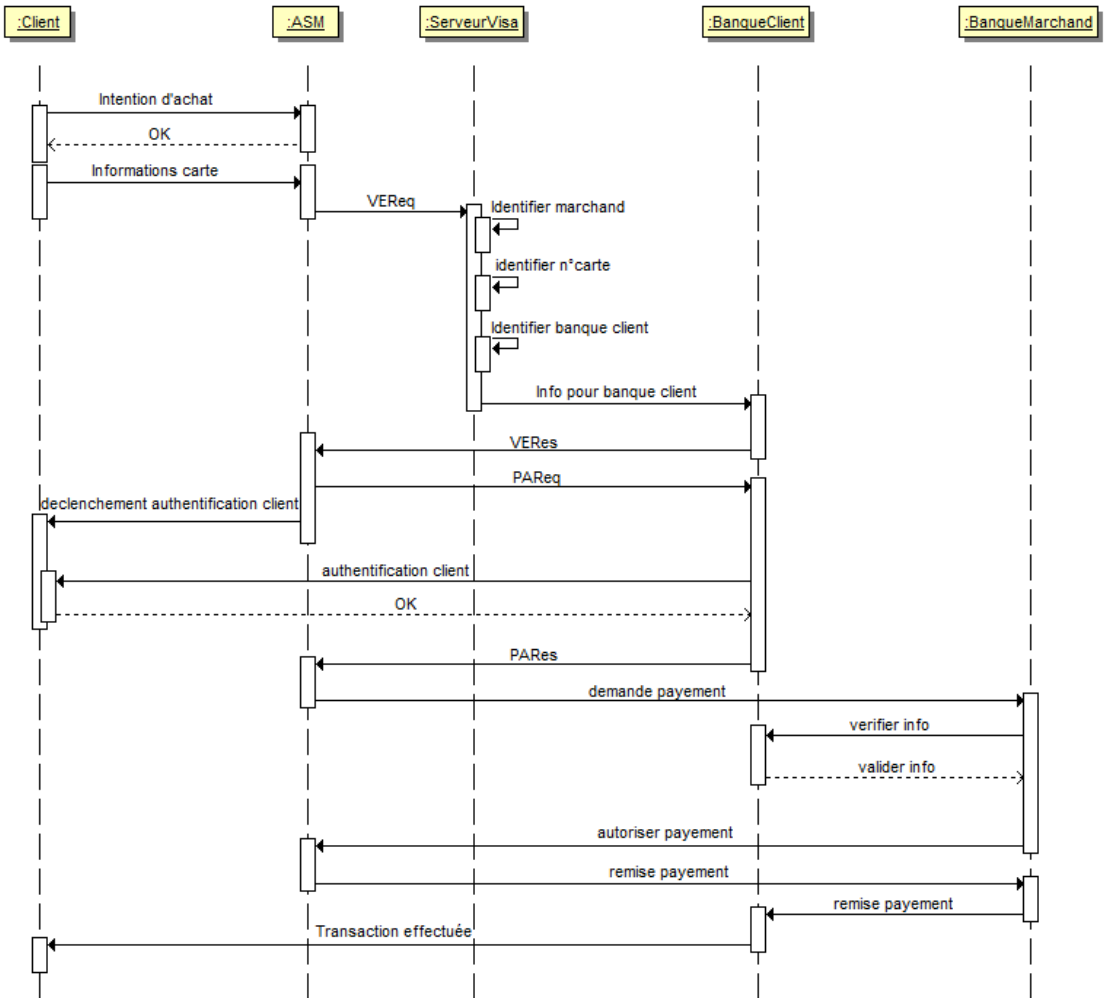


Fig 7 – Diagramme de séquence.

Le diagramme des cas d'utilisations, tout comme le diagramme de séquence, se base à la fois sur la théorie et sur les messages effectivement échangés observés en TP.

Nous avons ensuite réalisé le diagramme de classes suivant :

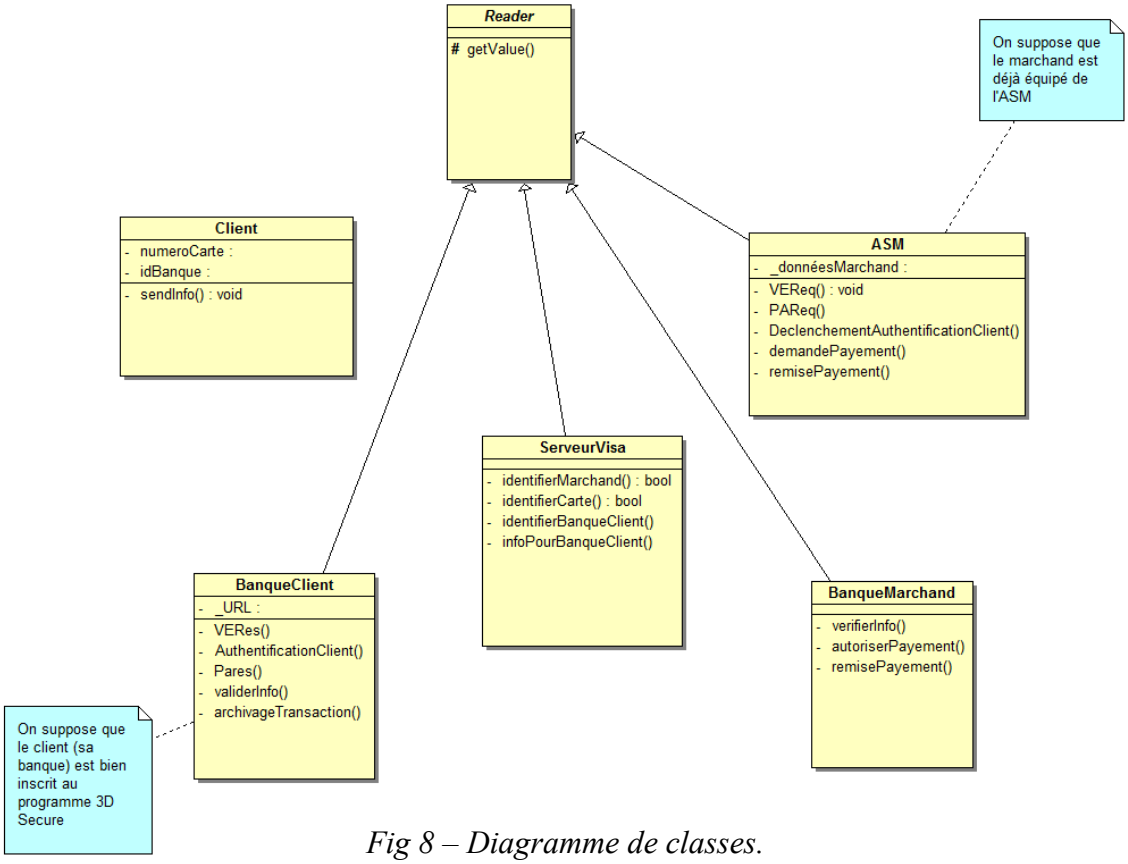


Fig 8 – Diagramme de classes.

Nous avons défini 5 classes principales correspondant aux différents acteurs de la transaction : Client, ASM, ServeurVisa, BanqueClient, BanqueMarchand.

Pour réaliser les classes en C#, nous avons utilisé le logiciel Visual Studio C Sharp Express 2010. Ce nouveau langage de programmation n'a pas posé de grandes difficultés, néanmoins la navigation dans la documentation proposée par Microsoft [10] n'a pas été des plus faciles. En effet, il n'est pas possible d'effectuer des recherches précises de méthodes simplement. Il faut d'abord trouver l'information du nom de la méthode ou de la classe capable de faire ce dont on a besoin, et c'est seulement ensuite que l'on peut aisément obtenir les classes mères ou dérivées. La documentation Java est sur ce point meilleure. Par contre, les classes et les méthodes sont souvent claires, détaillées et illustrées avec des exemples bien plus pertinents que sur la Javadoc.

En pratique, chaque classe envoie ses messages par génération de fichiers XML en utilisant les méthodes de la classe XmlWriter du paquet System.Xml. Ainsi, la méthode Create crée le fichier, la méthode WriteStartDocument pose l'entête et les méthodes WriteStartElement, WriteString, ... permettent de remplir le fichier très simplement.

Exemple pour le message VERes avec un XmlWriter writer :

```
writer.WriteStartDocument();
```

```
writer.WriteStartElement("ThreeDSecure");
writer.WriteStartElement("Message");
writer.WriteAttributeString("id", "1234567");
writer.WriteStartElement("VEReq");
writer.WriteStartElement("version");
```

... .

En ce qui concerne la réception des messages, elle se fait par la lecture des fichiers XML via la classe XmlTextReader du paquet System.Xml. Puisque la lecture d'une valeur d'un élément dans un fichier XML se fait toujours de la même manière, nous avons choisi de créer une interface Reader, accessible par toutes les classes ayant besoin de lire des données stockées dans un ficher XML. La recherche dans le fichier s'effectue avec les attributs NodeType, Name, Value et la méthode Read().

Beaucoup de méthodes renvoient systématiquement des booléens. En effet, le nombre de fonctionnalités que nous avons pu développer a été limité par le temps. Mais de telles fonctions sont facilement identifiables, modifiables et permettent donc d'intégrer facilement de nouvelles fonctionnalités.

Nous avons ensuite créé une interface web, grâce au logiciel Visual Studio Web Developer Express 2010. La technologie utilisée est l'ASP.NET. Elle nous permet de faire le lien entre nos classes en C# (.cs) et les pages web (.aspx). Nous avons pour le moment une page web avec un bouton de soumission qui prend le numéro de carte du client et qui propose d'afficher les fichiers XML générés.

La première difficulté, après avoir choisi le logiciel était de le comprendre ! Cela a nécessité beaucoup de recherches. Les vidéos et différents tutoriels proposés par Microsoft n'ont pas répondu à toutes nos interrogations. Il a fallut pratiquer le logiciel pendant un certain temps avant d'en comprendre les grands principes d'utilisation.

Visual Studio Web Developer propose un développement en mode graphique des pages Web avec une ToolBox contenant les principaux éléments utilisables. Ainsi, pour insérer un élément dans une page il suffit de glisser l'élément depuis la ToolBox. Ensuite, l'association des fonctions C# ou des évènements à un élément se fait dans ses propriétés (clic droit).

Les fichiers d'affichage web ont l'extension ".aspx" et, lorsque qu'ils sont créés dans le projet, ils sont automatiquement associés à un fichier ".aspx.cs" et ".designer.cs". Le premier permet d'utiliser des fonctions C# associés aux évènements des différents éléments de la page Web et le deuxième est mis à jour automatiquement pour les déclarations. Le projet contient aussi une page "maître" ".aspx" qui contient les élément statiques entre les différentes pages.

La dernière difficulté consistait à trouver le nom des fonctions permettant d'effectuer l'action désirée. Par exemple, pour accéder à une nouvelle page, la fonction à utiliser est :

```
Response.Redirect("Authentification.aspx");
```
(si l'on veut accèder à "Authentification.aspx")

Malgré le fait que le développeur soit guidé dans l'implémentation (les fonctions utilisables sont toutes répertoriées dans une fenêtre qui apparaît sur le curseur), il nous a fallu parfois les passer quasiment "une-par-une" jusqu'à trouver la bonne.

Au final, nous avons ainsi réalisé une application simple, sans préoccupations liées à la sécurité (les messages sont stockés dans des fichiers XML non protégés), mais qui permet de générer les messages d'un échanges 3D secure jusqu'à la demande de paiement du marchand. La principale difficulté pour l'implémentation résidait dans la découverte de Visual Studio et de l'ASP.NET qui ne sont pas au programme de l'école. Par ailleurs, ces recherches dans le cadre du projet sont pour nous très enrichissantes et représentent une bonne expérience puisque ces technologies sont utilisées dans de nombreux domaines professionnels.

III) Organisation du travail.

La section suivante concerne la méthodologie employée au cours du projet ainsi que la chronologie.

1 Méthodologie.

Le travail à accomplir au premier semestre était un travail de recherche de documentation, d'analyse et de synthèse, afin de comprendre le fonctionnement du protocole 3D secure que nous n'avions alors pas encore étudié en cours. La compréhension fut progressive tout au long du semestre, mais cela ne nous a pas posé de difficultés majeures.

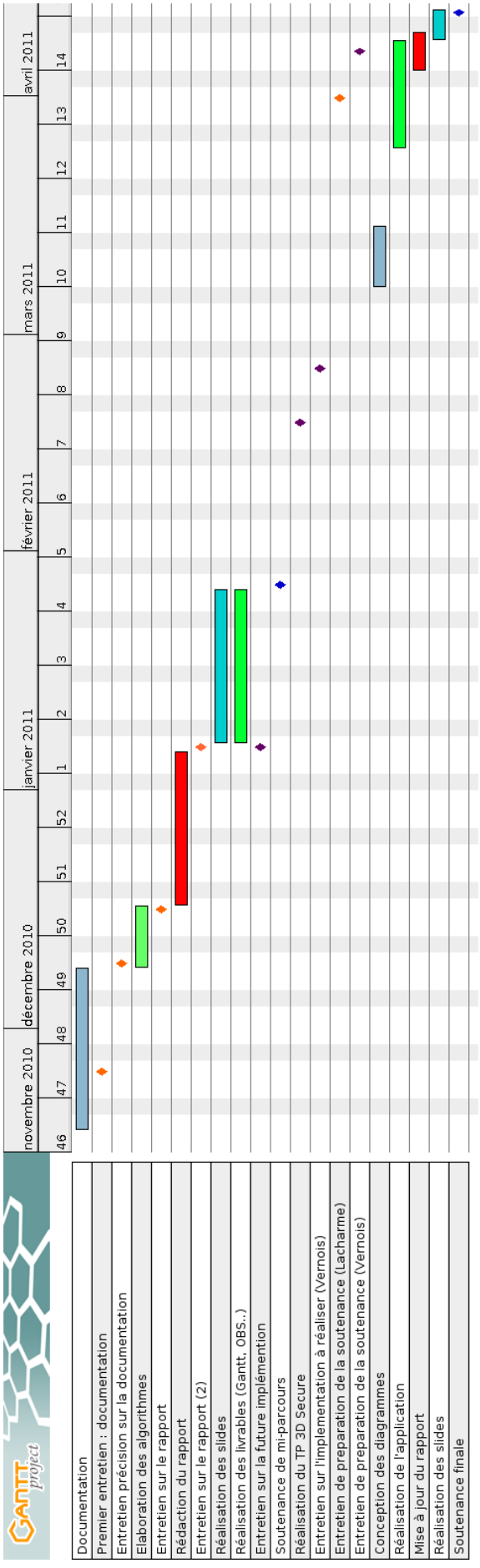
Lors du deuxième semestre, l'avancement s'est fait de manière beaucoup plus inégale. Les problèmes rencontrés sont mineurs, mais retardent à chaque fois l'avancement du projet.

En premier lieu, il y a eu un manque évident de temps, le deuxième semestre étant beaucoup plus court que le premier, et l'emploi du temps chargé. Nous avons d'abord dû trouver un créneau de libre pour effectuer le TP 3D secure. L'encadrant chargé de nous guider sur le TP et la partie implémentation a malheureusement eu quelques soucis de santé et a dû s'absenter plusieurs fois ce qui nous a fait mettre de coté le projet plusieurs semaines en février. De plus, il y a eu quelques cafouillages sur l'emploi du temps des deuxièmes années monétique, ce qui fait que tous les créneaux de l'emploi du temps susceptibles d'être libres pour le projet étaient occupés, mis à part pour la toute dernière semaine.

L'avancement a donc été inégal. Nous avons surtout progressé quelques semaines après le début du deuxième semestre, lorsque nous avons effectué le TP et que nous avons procédé à l'analyse de celui-ci, et lors de la fin du deuxième semestre, avec des créneaux se libérant sur notre emploi du temps.

2 Chronologie :

Fig 10 – Diagramme de Gantt (voir ci-contre) :



Conclusion

A l'heure actuelle, l'utilisation faite du protocole 3D secure peut être considérée comme fiable. Pourtant, les fraudes bancaires étant rares, l'utilisation de celui-ci n'est souvent pas rentable pour les marchands qui souhaiteraient l'appliquer.

Pour développer sa présence chez les sites marchands français, les banques ont donc mis en œuvre une stratégie de communication auprès de leurs clients et des marchands, en particulier depuis l'apparition du code à usage unique envoyé par SMS. Cette stratégie ne demande donc plus qu'à faire ses preuves en France. Au Royaume-Uni, les banques ont investi sur le protocole 3D secure dès son apparition, et aujourd'hui, si l'on regarde leurs 10 principaux sites marchands [9], plus de la moitié procèdent aux paiements avec 3D secure.

Concernant notre application, celle-ci est loin d'être terminée. Cependant, elle permet déjà de simuler un échange 3D-Secure et de générer les messages associés au format XML. L'application est également appelée à évoluer, et à intégrer de nombreuses fonctionnalités supplémentaires, comme des méthodes d'authentification du client différentes, ce qui pourrait permettre d'améliorer la maquette actuellement utilisée.

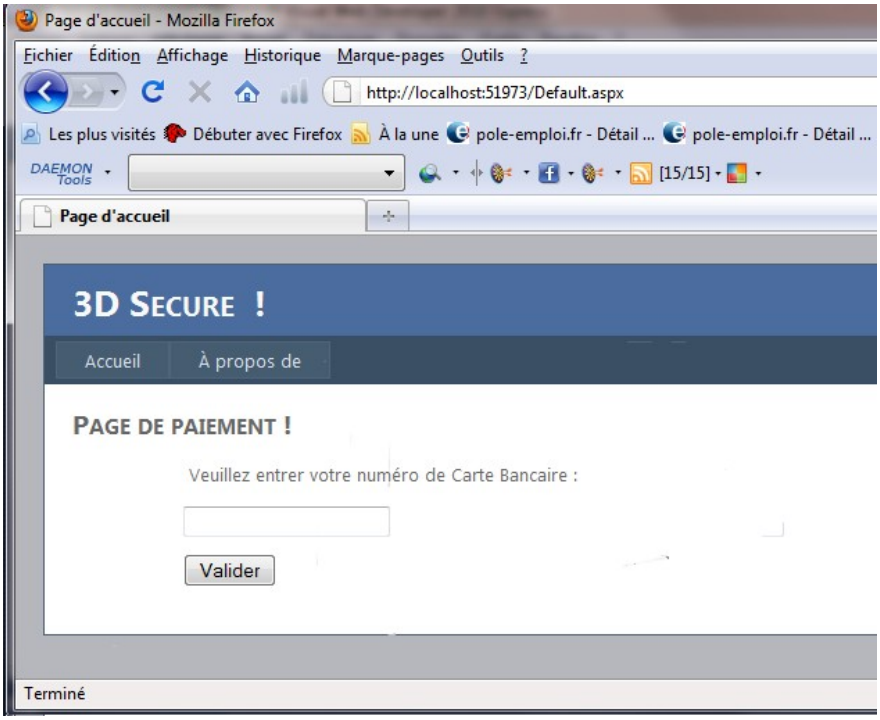


Fig 10 – Page d'accueil de notre application.

Références bibliographiques

[1] Mostafa Hashem Sherif. Paiements électroniques sécurisés. *Presses polytechniques et universitaires romandes*, 2007, 583 p.

[2] Marc Pasquet, Christophe Rosenberger, Félix Cuozzo. Security for Electronic Commerce. *Encyclopedia of Information Science and Technology, Second Edition,* 2009, 3383-3391 p.

[3] http://en.wikipedia.org/wiki/Public_key_infrastructure

[4] Giampaolo Bella1, Fabio Massacci, Lawrence C. Paulson, Piero Tramontano3. Making Sense of Specifications: the Formalization of SET . 2001, 8 p.

[5] Visa Corporation. 3D Secure Protocol Specification. 2004, 176 p.

[6] http://www.ecommerce404.fr/wp-content/uploads/2008/09/sips_3dsecure1.jpg

[7] Sites internet des principales banques. Ex :
https://particuliers.societegenerale.fr/votre_site/configuration_securite/zoom_3d.html
http://www.caisse-epargne.fr/evo-authentification-cb.aspx

[8] FEVAD (http://www.fevad.com)

[9] http://www.netimperative.com

[10] Documentation c# de Microsoft
http://msdn.microsoft.com/library/kx37x362

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Web;
using System.IO;
using System.Net;

namespace _3D_Secure
{
    class Main
    {
        public Main(String num)
        {

            Client cl = new Client(num, "222222");
            ASM marchand = new ASM();
            ServeurVisa serv = new ServeurVisa();
            BanqueClient bc = new BanqueClient();
            marchand.PAReq();
        }
    }
}
```



```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;
using System.Collections;
using System.Web;

namespace _3D_Secure
{
    public abstract class Reader
    {

        protected String getValue(String element, String file)
        {
            XmlTextReader xmlReader = new XmlTextReader(file);
            while (xmlReader.Read())
            {

                if ((xmlReader.NodeType == XmlNodeType.Element) && (element == x
mlReader.Name))
                {
                    while (xmlReader.Read())
                    {
                        if (xmlReader.NodeType == XmlNodeType.Text)
                        {
                            return xmlReader.Value;
                        }
                    }
                }
            }
            Console.WriteLine("Impossible de trouver l element : {0}", element);
            return "fail";
        }

    }
}
```

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;

namespace _3D_Secure
{
    class ASM : Reader
    {
        private String merID = "00008888";
        private String name = "3D Shop";
        private String country = "250";
        private String url = "www.3DShop.com";
        private String price = "242,00 euros";
        private String idBanqueClient=null;

        public ASM()
        {
            String pan = getValue("pan", "C:\\Users\\Baptiste\\Desktop\\3D Secure\\3D Secure\\inf
oClient.xml");
            String idBanqueClient = getValue("acqBIN", "C:\\Users\\Baptiste\\Desktop\\3D Se
cure\\3D Secure\\infoClient.xml");
            VEReq(pan, idBanqueClient);

        }


        void VEReq(String numeroCarte, String idBanque)
        {
            XmlWriterSettings settings = new XmlWriterSettings();
            settings.Indent = true;
            settings.IndentChars = ("  ");
            try
            {
                //on crÃ©e le fichier XML pour le message VEReq
                using (XmlWriter writer = XmlWriter.Create("C:\\Users\\Baptiste\\Desktop\
\3D Secure\\3D Secure\\VEReq.xml", settings))
                {
                    writer.WriteStartDocument();
                    writer.WriteStartElement("ThreeDSecure");
                    writer.WriteStartElement("Message");
                    writer.WriteAttributeString("id", "1234567");
                    writer.WriteStartElement("VEReq");
                    writer.WriteStartElement("version");
                    writer.WriteString("1.0.0");
                    writer.WriteEndElement();
                    writer.WriteStartElement("pan");
                    writer.WriteString(numeroCarte);
                    writer.WriteEndElement();
                    writer.WriteStartElement("Merchant");
                    writer.WriteStartElement("acqBIN");  //identificateur de la
banque du client
                    writer.WriteString(idBanque);
                    writer.WriteEndElement();
                    writer.WriteStartElement("merID");
                    writer.WriteString(merID);
                    writer.WriteEndElement();
                    writer.WriteStartElement("password");
                    writer.WriteString("azerty");
                    writer.WriteEndElement();
                    writer.WriteEndElement();
```



```
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception:{0}", e.ToString());
            }
        }



        public void PAReq()
        {
            XmlWriterSettings settings = new XmlWriterSettings();
            settings.Indent = true;
            settings.IndentChars = ("  ");
            try
            {
                //on crÃ©e le fichier XML pour le message PAReq
                using (XmlWriter writer = XmlWriter.Create("C:\\Users\\Baptiste\\Desktop\
\3D Secure\\3D Secure\\PAReq.xml", settings))
                {
                    writer.WriteStartDocument();
                    writer.WriteStartElement("ThreeDSecure");
                    writer.WriteStartElement("Message");
                    writer.WriteAttributeString("id", "1234567");
                    writer.WriteStartElement("PAReq");
                    writer.WriteStartElement("version");
                    writer.WriteString("1.0.0");
                    writer.WriteEndElement();
                    writer.WriteStartElement("Merchant");
                    writer.WriteStartElement("acqBIN");
                    writer.WriteString(idBanqueClient);
                    writer.WriteEndElement();
                    writer.WriteStartElement("merID");
                    writer.WriteString(merID);
                    writer.WriteEndElement();
                    writer.WriteStartElement("name");
                    writer.WriteString(name);
                    writer.WriteEndElement();
                    writer.WriteStartElement("country");
                    writer.WriteString(country);
                    writer.WriteEndElement();
                    writer.WriteStartElement("url");
                    writer.WriteString(url);
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteStartElement("purchase");
                    writer.WriteStartElement("price");
                    writer.WriteString(price);
                    writer.WriteEndElement();
                    writer.WriteStartElement("date");
                    writer.WriteString(DateTime.Now.ToString());
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                }
            }
            catch (Exception e)
            {
```

ANNEXES **ASM.cs** Page 3/3

```
                Console.WriteLine("Exception:{0}", e.ToString());
            }
        }
    }
}
```

ANNEXES **BanqueClient.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;

namespace _3D_Secure
{
    class BanqueClient : Reader
    {
        private String url = "www.EnsiBankB.fr";

        public BanqueClient()
        {
            VERes(url);
        }

        void VERes(String url){
            XmlWriterSettings settings = new XmlWriterSettings();
            settings.Indent = true;
            settings.IndentChars = ("  ");
            try
            {
                //on crÃ©e le fichier XML pour le message VERes
                using (XmlWriter writer = XmlWriter.Create("C:\\Users\\Baptiste\\Desktop\
\3D Secure\\3D Secure\\VERes.xml", settings))
                {
                    writer.WriteStartDocument();
                    writer.WriteStartElement("ThreeDSecure");
                    writer.WriteStartElement("Message");
                    writer.WriteAttributeString("id", "1234567");
                    writer.WriteStartElement("VERes");
                    writer.WriteStartElement("version");
                    writer.WriteString("1.0.0");
                    writer.WriteEndElement();
                    writer.WriteStartElement("url");
                    writer.WriteString(url);
                    writer.WriteEndElement();
                    writer.WriteStartElement("Protocol");
                    writer.WriteString("ThreeDSecure");
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception:{0}", e.ToString());
            }
        }
    }
}
```

ANNEXES **BanqueMarchand.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace _3D_Secure
{
    class BanqueMarchand : Reader
    {
    }
}
```

ANNEXES **Client.cs** Page 1/2

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml;

namespace _3D_Secure
{
    class Client
    {
        private String numeroCarte;
        private String idBanque;

        public Client(String nCard, String idB)
        {
            numeroCarte = nCard;
            idBanque = idB;

            sendInfo(numeroCarte, idBanque);


        }

        void sendInfo(String numeroCarte, String idBanque)
        {
            XmlWriterSettings settings = new XmlWriterSettings();
            settings.Indent = true;
            settings.IndentChars = ("  ");
            try
            {
                //on crÃ©e le fichier XML pour passer les infos du client au mar
chand
                using (XmlWriter writer = XmlWriter.Create("C:\\Users\\Baptiste\\Desktop\
\3D Secure\\3D Secure\\infoClient.xml", settings))
                {
                    writer.WriteStartDocument();
                    writer.WriteStartElement("ThreeDSecure");
                    writer.WriteStartElement("Message");
                    writer.WriteAttributeString("id", "1234567");
                    writer.WriteStartElement("infoClient");
                    writer.WriteStartElement("version");
                    writer.WriteString("1.0.0");
                    writer.WriteEndElement();
                    writer.WriteStartElement("Client");
                    writer.WriteStartElement("pan");
                    writer.WriteString(numeroCarte);
                    writer.WriteEndElement();
                    writer.WriteStartElement("acqBIN");  //identificateur de la
banque du client
                    writer.WriteString(idBanque);
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                    writer.WriteEndElement();
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("Exception:{0}", e.ToString());
            }
        }
```

```
    }
}
```



```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace _3D_Secure
{
    class ServeurVisa : Reader
    {
        public ServeurVisa()
        {
            if (identifierMarchand() && identifierCarte() && identifierClient())
 infoPourBanque();
        }

        bool identifierMarchand()
        {
            return true;
        }

        bool identifierCarte()
        {
            return true;
        }

        bool identifierClient()
        {
            return true;
        }

        void infoPourBanque()
        {

        }
    }
}
```

ANNEXES **About.aspx** Page 1/1

```
ï»¿<%@ Page Title="Qui sommes-nous" Language="C#" MasterPageFile="~/Site.master"
 AutoEventWireup="true"
    CodeBehind="About.aspx.cs" Inherits="_3D_Secure.About" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent
">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        Ã‰-^@ propos de 3D Secure
    </h2>
    <p>
        Ce projet a Ã©tÃ© rÃ©alisÃ© par Samuel AUGUSTE et Baptiste GIROUD en deu
xiÃ¨me annÃ©e MONETIQUE Ã  l'ENSICAEN
    </p>
</asp:Content>
```

ANNEXES **About.aspx.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace _3D_Secure
{
    public partial class About : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

ANNEXES **About.aspx.designer.cs** Page 1/1

```
ï»¿//------------------------------------------------------------------------------
---
// <gÃ©nÃ©rÃ© automatiquement>
//     Ce code a Ã©tÃ© gÃ©nÃ©rÃ© par un outil.
//
//     Les modifications apportÃ©es Ã  ce fichier peuvent provoquer un comportem
ent incorrect et seront perdues si
//     le code est rÃ©gÃ©nÃ©rÃ©.
// </gÃ©nÃ©rÃ© automatiquement>
//------------------------------------------------------------------------------

namespace _3D_Secure {


    public partial class About {
    }
}
```

ANNEXES **Authentification.aspx** Page 1/1

```
ï»¿<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireu
p="true" CodeBehind="Authentification.aspx.cs" Inherits="_3D_Secure.WebForm2" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <h2>Authentification DE ENSIBANK B</h2>
    <p>
        Entrez le code que nous vous avons communiquÃ© :    
  </p>
    <p>
        <asp:TextBox ID="TextBox1" runat="server" ontextchanged="TextBox1_TextCh
anged"></asp:TextBox>
    </p>
    <p>
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
            Text="Valider" />
         </p>
</asp:Content>
```

ANNEXES **Authentification.aspx.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace _3D_Secure
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Details.aspx");
        }

        protected void TextBox1_TextChanged(object sender, EventArgs e)
        {
            //mettre la suite de la transaction ici !!!

        }
    }
}
```

ANNEXES **Authentification.aspx.designer.cs** Page 1/1

```
ï»¿//------------------------------------------------------------------------------
// <gÃ©nÃ©rÃ© automatiquement>
//     Ce code a Ã©tÃ© gÃ©nÃ©rÃ© par un outil.
//
//     Les modifications apportÃ©es Ã  ce fichier peuvent provoquer un comportem
ent incorrect et seront perdues si
//     le code est rÃ©gÃ©nÃ©rÃ©.
// </gÃ©nÃ©rÃ© automatiquement>
//------------------------------------------------------------------------------

namespace _3D_Secure {


    public partial class WebForm2 {

        /// <summary>
        /// ContrÃ´le TextBox1.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.TextBox TextBox1;

        /// <summary>
        /// ContrÃ´le Button1.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button Button1;
    }
}
```

ANNEXES **Default.aspx** Page 1/1

```
ï»¿<%@ Page Title="Page d'accueil" Language="C#" MasterPageFile="~/Site.master"
AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="_3D_Secure._Default" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent
">
</asp:Content>
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <h2>
        Page de paiement !</h2>
    <p>
         Veuillez entrer votre numÃ©ro de Carte Bancaire :   
;   </p>
    <p>
        <asp:TextBox ID="TextBox1" runat="server" ontextchanged="TextBox1_TextCh
anged"></asp:TextBox>
    </p>
    <p>
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
            Text="Valider" />
         </p>



</asp:Content>
```

ANNEXES **Default.aspx.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace _3D_Secure
{
    public partial class _Default1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Authentification.aspx");
        }

        protected void TextBox1_TextChanged(object sender, EventArgs e)
        {
            Main main = new Main(TextBox1.Text);

        }

    }
}
```

ANNEXES **Default.aspx.designer.cs** Page 1/1

```
ï»¿//------------------------------------------------------------------------------
---
// <gÃ©nÃ©rÃ© automatiquement>
//     Ce code a Ã©tÃ© gÃ©nÃ©rÃ© par un outil.
//
//     Les modifications apportÃ©es Ã  ce fichier peuvent provoquer un comportem
ent incorrect et seront perdues si
//     le code est rÃ©gÃ©nÃ©rÃ©.
// </gÃ©nÃ©rÃ© automatiquement>
//------------------------------------------------------------------------------

namespace _3D_Secure {


    public partial class _Default {

        /// <summary>
        /// ContrÃ´le TextBox1.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.TextBox TextBox1;

        /// <summary>
        /// ContrÃ´le Button1.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button Button1;
    }
}
```

ANNEXES **Details.aspx** Page 1/1

```
ï»¿<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master" AutoEventWireu
p="true" CodeBehind="Details.aspx.cs" Inherits="_3D_Secure.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeadContent" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="MainContent" runat="server">
    <h2>DÃ©tails de la transaction </h2>
    <p> </p>
    <p>
        <asp:Button ID="Button1" runat="server" onclick="Button1_Click"
            Text="infoClient.xml" />
    </p>
    <p>
        <asp:Button ID="Button2" runat="server" onclick="Button2_Click"
            style="height: 26px" Text="VEReq.xml" />
    </p>
    <p>
        <asp:Button ID="Button3" runat="server" onclick="Button3_Click"
            Text="VERes.xml" />
    </p>
    <p>
        <asp:Button ID="Button4" runat="server" onclick="Button4_Click"
            Text="PAReq.xml" />
    </p>
    <p> </p>
</asp:Content>
```

ANNEXES **Details.aspx.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace _3D_Secure
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("infoClient.xml");
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            Response.Redirect("VEReq.xml");
        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            Response.Redirect("VERes.xml");
        }

        protected void Button4_Click(object sender, EventArgs e)
        {
            Response.Redirect("PAReq.xml");
        }


    }
}
```

ANNEXES **Details.aspx.designer.cs** Page 1/1

```
ï»¿//------------------------------------------------------------------------------
---
// <gÃ©nÃ©rÃ© automatiquement>
//     Ce code a Ã©tÃ© gÃ©nÃ©rÃ© par un outil.
//
//     Les modifications apportÃ©es Ã  ce fichier peuvent provoquer un comportem
ent incorrect et seront perdues si
//     le code est rÃ©gÃ©nÃ©rÃ©.
// </gÃ©nÃ©rÃ© automatiquement>
//------------------------------------------------------------------------------

namespace _3D_Secure {


    public partial class WebForm1 {

        /// <summary>
        /// ContrÃ´le Button1.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button Button1;

        /// <summary>
        /// ContrÃ´le Button2.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button Button2;

        /// <summary>
        /// ContrÃ´le Button3.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button Button3;

        /// <summary>
        /// ContrÃ´le Button4.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Button Button4;
    }
}
```

ANNEXES **Site.Master** Page 1/1

```
ï»¿<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site.master.cs" I
nherits="_3D_Secure.SiteMaster" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/x
html1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title></title>
    <link href="~/Styles/Site.css" rel="stylesheet" type="text/css" />
    <asp:ContentPlaceHolder ID="HeadContent" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form runat="server">
    <div class="page">
        <div class="header">
            <div class="title">
                <h1>
                    3D Secure  !
                </h1>
            </div>
            <div class="loginDisplay">
            </div>
            <div class="clear hideSkiplink">
                <asp:Menu ID="NavigationMenu" runat="server" CssClass="menu"
                    EnableViewState="false" IncludeStyleBlock="false" Orientatio
n="Horizontal"
                    onmenuitemclick="NavigationMenu_MenuItemClick">
                    <Items>
                        <asp:MenuItem NavigateUrl="~/Default.aspx" Text="Accueil
"/>
                        <asp:MenuItem NavigateUrl="~/About.aspx" Text="Ã‰-^@ pro
pos de"/>
                    </Items>
                </asp:Menu>
            </div>
        </div>
        <div class="main">
            <asp:ContentPlaceHolder ID="MainContent" runat="server"/>
        </div>
        <div class="clear">
        </div>
    </div>
    <div class="footer">

    </div>
    </form>
</body>
</html>
```

ANNEXES **Site.Master.cs** Page 1/1

```
ï»¿using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace _3D_Secure
{
    public partial class SiteMaster : System.Web.UI.MasterPage
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void HeadLoginView_ViewChanged(object sender, EventArgs e)
        {

        }

        protected void NavigationMenu_MenuItemClick(object sender, MenuEventArgs
 e)
        {

        }
    }
}
```

```
ï»¿//------------------------------------------------------------------------------
---
// <gÃ©nÃ©rÃ© automatiquement>
//     Ce code a Ã©tÃ© gÃ©nÃ©rÃ© par un outil.
//
//     Les modifications apportÃ©es Ã  ce fichier peuvent provoquer un comportem
ent incorrect et seront perdues si
//     le code est rÃ©gÃ©nÃ©rÃ©.
// </gÃ©nÃ©rÃ© automatiquement>
//------------------------------------------------------------------------------

namespace _3D_Secure {


    public partial class SiteMaster {

        /// <summary>
        /// ContrÃ´le HeadContent.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.ContentPlaceHolder HeadConte
nt;

        /// <summary>
        /// ContrÃ´le HeadLoginView.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.LoginView HeadLoginView;

        /// <summary>
        /// ContrÃ´le NavigationMenu.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.Menu NavigationMenu;

        /// <summary>
        /// ContrÃ´le MainContent.
        /// </summary>
        /// <remarks>
        /// Champ gÃ©nÃ©rÃ© automatiquement.
        /// Pour modifier, dÃ©placez la dÃ©claration de champ du fichier de conc
epteur dans le fichier code-behind.
        /// </remarks>
        protected global::System.Web.UI.WebControls.ContentPlaceHolder MainConte
nt;
    }
}
```