République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahleb Blida - Blida 1

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'étude pour l'obtention
d'un Master2 en Informatique
option SSI et SIR

Thème :

«Attaques physiques et logiques ciblant la carte à puce»

Réalisé par :

MAHDI Khaldi

LAHMER Benhaoua

Date : 04/06/2023

Membres de jury :

Mme Ykhlef Hadjer	Président
Mme Gheghoub Yasmina	Examinateur
Mr. KAMECHE Abdallah	Promoteur

Remerciements

Tout d'abord, Nous remercions ***ALLAH*** *le tout puissant d'avoir nous donner le courage, la volonté et la patience de mener à terme le présent travail.*

Nous tenons à remercier dans un premier temps, toute l'équipe Pédagogique et administrative de l'Université Saad Dahleb Blida - Blida 1 ainsi que les intervenants professionnels responsables de la formation master2 sécurité des systèmes d'information SSI et master2 Systèmes Informatiques et Réseaux SIR.

Avant d'entamer ce rapport, nous profitons de l'occasion pour remercier le Chef du Département Informatique Mr. Oueld AISSA Ahmed, le promoteur Mr. KAMECHE Abdellah et l'équipe pédagogique qui n'ont pas cessé de nous encourager pendant la durée de la formation, ainsi pour leurs générosités en matière de formation et d'encadrement.

Nous les remercions également pour l'aide et les conseils concernant les missions évoquées dans ce rapport, qu'ils nous ont apporté lors des différents suivis, et la confiance qu'ils nous ont témoigné.

Grand Merci à tout le monde.

Résumé

Les cartes à puce sont considérées comme le moyen de prédilection dans les pays où le commerce électronique (paiement de proximité ou à distance) fait partie intégrante de la vie quotidienne des citoyens. Elles sont un moyen fiable, mais nécessitent des précautions minimales et des mesures de sécurité pour assurer des transactions plus sûres et sécurisées, afin de contrer les fraudes et les attaques visant ces moyens électroniques.

Cependant, les techniques d'attaques et de fraudes visant les cartes à puce ont connu une augmentation considérable dans les pays qui ont adopté l'utilisation des cartes à puce comme moyen de paiement électronique. Dans ce contexte, et étant donné que l'Algérie a déployé ces dernières années des solutions monétiques électroniques basées sur les cartes à puce, notre objectif est de comprendre les types d'attaques et de menaces qui peuvent cibler les cartes à puce, en particulier dans le domaine monétique.

On s'intéresse ici à mettre en évidence les vulnérabilités présentes dans les plateformes monétiques utilisées dans le pays et en particulier celles liées à l'utilisation du protocole 3DSecure. Nous avons conçu et mis en œuvre plusieurs attaques, en simulant différents scénarios. Grâce à ces simulations, nous avons pu identifier certaines failles au sein des plateformes monétiques algériennes.

Mots clés : Carte à puce, Sécurité, 3DSecure, Attaques , Algérie, Vulnérabilités

Abstract

Smart cards are considered the preferred method of payment in countries where electronic commerce (both in-person and remote) is an integral part of citizens' daily lives. They are a reliable means of payment, but they require minimal precautions and security measures to ensure safer and more secure transactions, countering fraud and attacks targeting these electronic methods.

However, techniques for attacking and defrauding smart cards have significantly increased in countries that have adopted them as a means of electronic payment. In this context, given that Algeria has deployed electronic monetary solutions based on smart cards in recent years, our objective is to understand the types of attacks and threats that can target smart cards, particularly in the monetary domain.

Here, we aim to highlight the vulnerabilities present in the monetary platforms used in the country, specifically those related to the use of the 3DSecure protocol. We have designed and implemented several attacks, simulating different scenarios. Through these simulations, we have been able to identify certain flaws within Algerian monetary platforms.

Keywords: Smart cards, Security, 3DSecure, Attacks, Algeria , Vulnerabilities

الملخص

البطاقات الذكية تُعتبر وسيلة الدفع المفضلة في البلدان التي يعد التجارة الإلكترونية (الدفع المباشر أو عن بُعد) جزءًا لا يتجزأ من حياة المواطنين اليومية. إنها وسيلة موثوقة، ولكنها تتطلب احتياطات دنيا وتدابير أمنية لضمان إجراءات تحويل أكثر أمانًا وأمانًا، ولمواجهة عمليات الاحتيال والهجمات التي تستهدف هذه الوسائل الإلكترونية.

ومع ذلك، فقد شهدت تقنيات الهجمات والاحتيال المستهدفة للبطاقات الذكية زيادة ملحوظة في البلدان التي اعتمدت استخدام البطاقات الذكية كوسيلة للدفع الإلكتروني. في هذا السياق، ونظرًا لأن الجزائر نفذت في السنوات الأخيرة حلولًا نقدية إلكترونية مبنية على البطاقات الذكية، فإن هدفنا هو فهم أنواع الهجمات والتهديدات التي يمكن أن تستهدف البطاقات الذكية، خاصة في المجال المالي.

نحن هنا نسعى لتسليط الضوء على الثغرات الموجودة في المنصات المالية المستخدمة في البلاد، وعلى وجه الخصوص تلك المرتبطة باستخدام بروتوكول 3DSecure
لقد قمنا بتصميم وتنفيذ العديد من الهجمات، محاكاة سيناريوهات مختلفة. من خلال هذه المحاكاة، تمكنا من تحديد بعض الثغرات في المنصات المالية الجزائرية.

كلمات مفتاحية: البطاقات الذكية، الأمان، 3DSecure، الهجمات، الجزائر

Liste des figures

Liste des abréviations

APDU	*Application Protocol Data Unit*
AID	*Application IDentifer*
API	*Application Programming Interface*
ATR	*Answer To Reset*
CAD	*Card Acceptance Device*
CAP	*Converted Applet*
CISC	*Complex Instruction Set Computer*
COS	*Card/Chip Operating System*
CPU	*Central Processing Unit*
EEPROM	*Electrecally, Erasable, Programmable Read Only Memory*
HTTP	*Hyper Text Transmission Protocol*
HTML	*Hyper Text Markup Language*
ISO	*International Standards Organization*
JAVA EE	*Java Entreprise Edition*
JCRE	*Java Card Runtime Environment*
JCVM	*Java Card Virtual Machine*
JVM	*Java Virtual Machine*
PIN	*Personal Identification Number*
PIX	*Proprietary Identifier eXtension*
RAM	*Random Access Memory*
RNG	*Random Number Generator*
RID	*Ressource Identifier*
CSS	*Feuilles de Style en Cascade*
ROM	*Read Only Memory*
SIM	*Subscriber Identity Module*
SW	*Status Word*
TAPDU	*Transport Application Protocol Data Unit*
SPA	*Simple Power Attack*
DPA	*Differential Power Attack*
PPS	*protocol parameter selection*
DSP2	*Directive Européenne Relative Aux Services De Paiement*
CIB	*Carte Inter Bancaire*
SATIM	*Société d'Automatisation des Transactions Interbancaires et de Monétique*
DNS	*Domaine Name System*

Sommaire

Introduction générale

- **Mise en contexte**

L'internet s'est progressivement transformé en un canal de distribution électronique sur lequel les entreprises, les commerçants, les consommateurs et les administrations échangent, commercialisent des biens et des services. Au niveau mondial, cette transformation de l'internet en un espace économique, commerciale, voire même administratif, a cependant été très rapide et constitue désormais et depuis plusieurs années un espace de transfert de donnée incontournable.

Dans ce contexte, plusieurs solutions ont été déployées pour garantir la sécurité de l'information et sa protection durant l'exécution des transactions électroniques. L'utilisation des cartes à puce est l'un de ces solutions qui a été largement exploitée au niveau mondial, et ce, pour garantir un certain niveau de sécurité. En Algérie, ces dernières années, plusieurs solutions ont été conçues et déployées pour l'utilisation des services électronique au profit des citoyens qui utilisent les cartes à puces, tels que :

- Le développement du paiement électronique, notamment grâces aux solutions mises en place par la Société d'Automatisation des Transactions Interbancaires et de Monétique (SATIM), Algérie Poste, le Groupement d'Intérêt Economique Monétique (GIE-Monétique), la Banque d'Algérie (BA). (Utilisation des cartes à puce).
- L'utilisation de la carte nationale d'identité biométrique électronique (CNIBE) et le passeport biométrique depuis 2016 par les services relevant du ministère de l'intérieure. (Utilisation des cartes à puce).
- L'utilisation de la carte d'assurance Chiffa par la caisse nationale des assurances sociales (CNAS). (Utilisation des cartes à puce), etc.

- **Problématique**

Au niveau mondial, ces cartes à puce ont été largement exploitées dans plusieurs domaines (Bancaire, Santé, Télécom, Administration, Etc.). Dans le domaine bancaire et financier, la carte à puce a été utilisée comme un moyen de paiement préféré dans les pays où le commerce électronique (paiement à proximité ou à distance) fait partie de la vie quotidienne des citoyens. C'est un moyen de paiement fiable qui nécessite cependant un minimum de précautions et des mesures de sécurité pour accomplir des transactions plus sûres et plus sécurisées contre les fraudeset les attaques qui ciblent ces moyens électroniques. Ces techniques d'attaques et de fraudes contre les cartes à puce ont connu une énorme évolution au niveau de ces pays. (*Statistiques ; Figure 1*). *[01]*

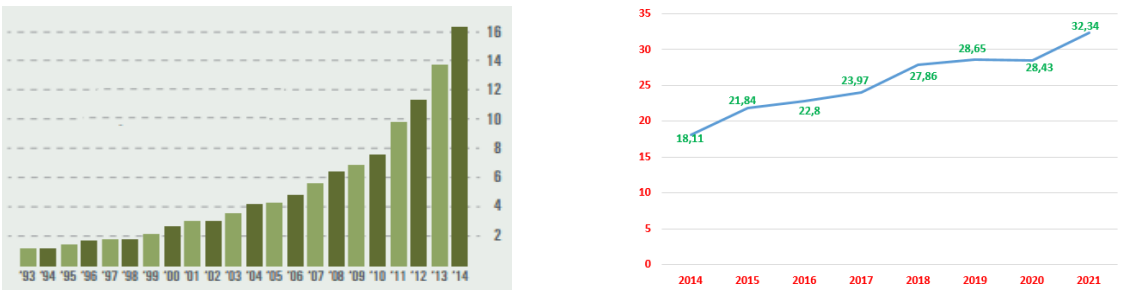


Figure 1- *Évolution de la fraude à la carte Bancaire dans le monde (1993-2021) en M$ [01]*

A l'instar des pays qui ont adopté l'utilisation de ces cartes pour sécuriser des transactions électroniques, notamment dans le domaine bancaire et financières, l'Algérie a déployé ces dernières années des solutions monétiques électroniques basées sur l'utilisation des cartes à puce,

à savoirs la carte EDAHABIA d'Algérie poste et la carte CIB émise par les différentes banques algériennes et étrangères installées au niveau national. Ces cartes servent à faire et sécuriser plusieurs transactions financières électroniques comme les achats, les paiements en ligne, le paiement de proximité via un terminal de paiement (TPE), ainsi destransactions financières via les distributeurs ou guichets automatiques de billets (DAB), de carburant, autoroute, etc.

Dans le contexte de l'Algérie, où l'activité de paiement électronique a connu une augmentation significative, il est essentiel de s'interroger sur le niveau de sécurité de ces transactions. Avec le déploiement de solutions monétiques électroniques basées sur l'utilisation de cartes à puce, il est important d'évaluer les mesures de sécurité mises en place pour protéger les données personnelles et financières des utilisateurs.

- **Objectif du travail :**

Le travail ciblé de notre part comme projet de fin d'étude, s'articule autour de la sécurité et les menaces qui ciblent les cartes à puce, les systèmes informatiques embarqués sur ces types de support, ainsi que les données qui existent au niveau de ces cartes. Ces dernières sont perçues comme un support d'exécution d'applications sécurisées capables de stocker, traiter et de manipuler des données sensibles de façon fiable, raison pour lesquelles elles sont de plus en plus utilisées à des besoins de sécurité.

L'objectif est de cerner et de comprendre quelques typologies d'attaques et de menaces qui peuvent cibler les cartes à puce et les plateformes monétiques en Algérie, passant par la fiabilitédes mécanismes et composantes de la sécurité qui ont été mises en place.

Ce travail se concrétisera par une implémentation et simulation d'éventuelles attaques ciblant les données de ces cartes, notamment lors d'une transaction de paiement en ligne (paiement électronique).

- **Organisation du mémoire**

Le mémoire est composé d'une introduction générale et trois (03) chapitres :

- Le premier chapitre présente les cartes à puce ainsi que les attaques qui ciblent ces dernières. Ce chapitre est divisé en deux parties sous forme de sections. La première section traite l'état de l'art des cartes à puce en décrivant leurs composants physiques, environnement logiciel, ainsi que l'aspect de sécurité qui y est lié. La deuxième section de ce chapitre traite les types d'attaques qui prennent en ciblece type de cartes, en mettant l'accent sur les techniques de réalisation de ces attaques.
- Le deuxième chapitre est consacré à la conception des scénarios d'attaques qui cible le vol et utilisation frauduleuse des données d'une carte à puce bancaire, lors d'une transaction en ligne.
- Le troisième chapitre est consacré à l'implémentation des simulations et la discussion autour de ses résultats.

Ce mémoire se termine par une conclusion générale

CHAPITRE I
Attaques sur la carte à puce
(Partie Théorique)

Chapitre I : Attaques sur la carte à puce

Introduction

Ce chapitre est divisé en deux sections. La première section est dédiée à la présentation des cartes à puce en décrivant leurs composantes physiques et leurs caractéristiques mécaniques et électriques, en passant par l'aspect sécurité de ce dernier. On mettra en évidence les mécanismes mis en place pourprotéger l'intégrité des cartes. Durant cette partie on va aborder aussi la couche applicative déployée dans ces cartes et le mécanisme de communication avec l'extérieur, notamment avecles lecteurs de cartes souvent appelés CAD pour Card Acceptance Device. La deuxième section est consacrée aux différentes attaques qui ciblent l'intégrité et la confidentialité des données et composantes des cartes à puce. Dans cette partie on va discuter les techniques qui sont largement exploitées pour contourner la sécurité des cartes soit en prenant le matériel en défaut, soit en prenant la couche applicative ou le système en défaut. Le travail achevé nous permet de cerner et de comprendre quelques typologies d'attaques et de menaces qui peuvent cibler les cartes à puce, en particulier dans le domaine monétique.

Section 1 : La carte à puce et son comportement

A travers cette section, nous allons présenter quelques concepts de base liés à la carte à puce.

1.1.C'est quoi une carte à puce

Une carte à puce, comme son nom l'indique, est une carte fabriquée en matière plastique (*standardisée ; norme ISO/IEC 7816*) *[01]*, de quelques centimètres de côté et moins d'un millimètred'épaisseur, et qui porte au moins un circuit intégré (une puce) capable de contenir et/ou de traiter de l'information (*à mémoire simple ou à microprocesseur*). Le circuit intégré peut agir autant qu'un support de l'information, comme il peut contenir un microprocesseur capable de traiter cette information. *[02]*

Les cartes à puce sont généralement utilisées dans plusieurs domaines à savoir : la monétique (*carte bancaire de débit ou de crédit, porte-monnaie électronique*), comme moyen d'identification personnelle (*carte d'identité biométrique, badge d'accès aux bâtiments, passeport*), la téléphonie mobile (*carte SIM*), secteur médicale (*carte d'assurance maladie*), la sécurité informatique (*Authentification forte et signature électronique, etc.*). *[03]*

1.2.Les différents types des cartes à puces

En prenant en compte les fonctionnalités de leurs circuits intégrés internes, les cartes à puce se distinguent surtout les unes des autres en deux grandes types et catégories de cartes(figure 2), ontrouve les cartes avec contact, et les cartes à puce sans contact (*Contactless*) et dans certain casmixtes (*Hybride). [04]*

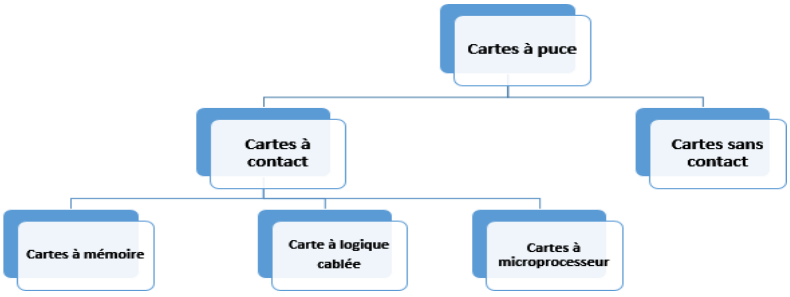


Figure 2- *Types de cartes à puce. [04]*

1.2.1. Cartes sans contact

En utilisant les cartes sans contacts, la communication avec le monde extérieur s'établit à travers une antenne imprimée ou intégrée dans la carte à puce et une interface radio (*réseau*

sans fil). La carte sans contact contient une puce électronique avec un émetteur hyperfréquence et une antenne intégrée dans le plastique. La carte doit être assez près du lecteur, entre 3 et 10 centimètres pour transmettre et réceptionner les données. ***Dans ce travail on va concentrer notre étude et recherches sur les cartes à contact, et ce, en vue que ce type de carte c'est le plus utilisées par les Banques algériennes et Algérie poste****. [03]*

1.2.2. Cartes à contact

On peut distinguer les grandes familles de cartes à puce à contact : Les cartes à mémoire simple ; les cartes à logique câblée et les cartes à microprocesseur. *[04][05]*

- **Les cartes à mémoire simple**

Les cartes à mémoire simple ne contiennent pas d'un microprocesseur intégré et ne peuvent pas traiter de l'information qu'elles contiennent. Ces types de cartes servent simplement d'unité de stockage (*Stocke des données*). Les données déjà présentes sont traitées par des fonctions simples préprogrammées sur la puce (*circuits préprogrammés*). *[04]*

- **Les cartes mémoire à logique câblée**

Avec les cartes mémoire à logique câblée, l'accès peut être restreint à un niveau prédéfini et déterminé via des circuits préprogrammés (*non reprogrammables*) et figés pour une application particulière, comme la vérification d'un code PIN.

Pour les besoins de flexibilité, la logique câblée a été remplacée par un microcontrôleur contenant un système d'exploitation capable de charger le programme à exécuter. Le système doit donc offrir des abstractions à l'application, assurer la confidentialité et l'intégrité des secrets, gérer les transmissions avec le lecteur et contrôler l'exécution du code. D'où la naissance des nouvelles cartes à puce à microprocesseur. *[04]*

- **Les cartes à microprocesseur**

Les cartes à microprocesseur ou smart carte contient un processeur et de la mémoire, le tout dans un microcontrôleur encarté. Ces dispositifs permettent à la carte d'être programmée pour exécuter un ou plusieurs types d'applications et collabore avec d'autres systèmes (*Applications Hot*) par l'intermédiaire des lecteurs généralement appelé CAD (*Card Acceptance Device*). Avec les cartes à microprocesseurs, on assiste à un véritable saut technologique et fonctionnel. Elles ont vraiment apporté des innovations majeures et extraordinaires, qui ont permet une flexibilité pratique et opérationnelle. Ce sont ces types de cartes qui ont permis de réaliser des systèmes d'identification et d'authentification offrant un très grand niveau de sécurité. *[05][04][06]*

1.3.Caractéristiques des cartes à puce (ISO-7816).

Dans les technologies des cartes à puce, la notion de la normalisation est très importante. En effet, un émetteur peut être réticent à investir dans une technologie potentiellement incompatible avec les générations futures. La norme ISO-7816 (1-15) est la norme de référence la plus connue pour les cartes à puce. Les quatre principales normes sont : ISO 7816-1, ISO 7816-2, ISO 7816-3, ISO 7816-4, elles spécifient entre autres les éléments et caractéristiques physiques et dimensionnels des supports plastiques, la résistance, la signification et la localisation des contacts, les protocoles et contenus des messages de bas et hauts niveaux échangés avec les cartes. Cette normalisation permet aux cartes à puces de fonctionner partout dans le monde et sur n'importes quel lecteur normalisé. *[04][07]*

1.3.1. Caractéristiques mécaniques

Dans le domaine des cartes à puce, généralement deux formats de cartes à puce sont les plus connus et les plus utilisés, celui de la carte bancaire (*Crédit, Débit, Paiement, etc*.) et celui de la carte SIM (*La Téléphonie Mobile*). Dans ce contexte, trois (3) formats normalisés existent :

ID1, ID00 et ID000 (Figure 3) . En termes de construction et fabrication, généralement, le fabricant produitune seule taille (ID1), le client final pourra réduire ses dimensions au format ID00 ou ID000 (*ex. carte SIM*). *[04]*

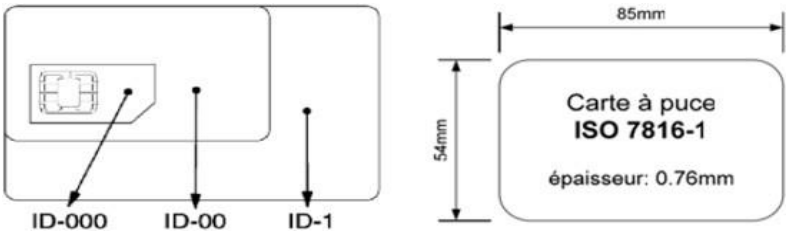


Figure 3- *Les dimensions des trois (3) formes de cartes à contact. [04]*

1.3.2. Les contacts des cartes à puce

Pour communiquer avec le monde extérieur, la carte à contact doit être insérée dans un lecteur, appelé aussi CAD, existe souvent sur plusieurs types d'appareils comme un terminale de paiement TPE, un distributeur automatique de billet DAB, un guichet automatique de billet GAB, etc. Suite la norme ISO 7816-2, cette carte possède des contacts présents sur sa surface afin qu'elle puisse communiquer avec ce lecteur. Les contacts sont en nombre de huit (8) (C1-C8) et dont les caractéristiques sont définies dans ladite norme. Le tableau suivant résume le fonctionnement des contacts.

Contact	Appellation	Utilisation
C1	**VCC**	(*Volt en Courant Continu*) est le point d'alimentation. La tension varie en général de 3 à 5 volts. (*la source de l'alimentation est le CAD*).
C2	**RST**	(*ReSeT*) utilisé pour réinitialiser le microprocesseur. Cela permet une "réinitialisation Chaude" et évite d'avoir à sortir et réintroduire la carte dans l'appareil en cas d'erreur (*réinitialisation froide*).
C3	**CLK**	Le processeur ne possédant pas d'horloge interne, celle-ci est située sur le point CLK (*CLocK*).
C5	**GND**	(*GrouND*) est la masse, sa tension est zéro.
C6	**Vpp**	(*Peak-Point Voltage*) est optionnel et n'est plus utilisé sur les cartes actuelles. Il permettait de contrôler la tension d'entrée.
C7	**I/O**	(*Input/Output*) est le point d'entrée-sortie permettant de transmettre les données à l'extérieur en mode semi-duplex : un seul sens de communication est autorisé à un moment donné.
C4 et C8	**RFU**	A l'origine, ces points de contact sont réservés pour utilisation future mais actuellement ils servent à communiquer en USB.

Tab 1 Fonctionnement des points de contacts d'une carte à puce. *. [05]*

1.3.3. Architecture d'une carte à microprocesseur

Pour assurer son fonctionnement de calcule et de stockage d'information, une carte à microprocesseur contient les composantes suivantes: *[08]*

- **ROM** (*Read Only Memory*) : Cette mémoire contient le système d'exploitation (*MultOS, Java Card, etc.*).
- **RAM** (*Random Access Memory*) : Cette mémoire est utilisée comme espace temporaire pour modifier et stocker les données, notamment pendant l'exécution d'une application et/ou la communication avec un CAD.

- **EEPROM** (*Electrical Erasable Programmable Read Only Memory*) **:** Cette mémoire stocke les applets à exécuter sur la Smart Card. Ces applets ne sont pas effacées et les données sont donc réutilisables dans le même état à chaque utilisation de la carte.
- **Microprocesseur** : Le processeur implanté dans les cartes à puce est très faible par rapport aux processeurs implantés dans un PC. Ces derniers ne contiennent qu'un jeu d'instructions limité. Il est conçu spécialement pour répondre aux systèmes embarqués sur carte à puce.
- **Crypto-Processeur** : La technologie des cartes à puce a utilisé, de plus en plus, un second crypto processeur sécurisé, adjoint au microprocesseur traditionnel de la carte à puce. Ce processeur accélère les grands calculs liés à la cryptographie. Ainsi, cette technologie a implanté un générateur de nombres aléatoires interne pour tirer les aléas nécessaires à tous les algorithmes cryptographiques.
- **Bus :** Les Bus sert à transférer de l'information entre les différentes composantes de la carte à puce, les bus se trouvent à la périphérie des zones mémoires, facilement reconnaissables sur les composants.

La figure 4 illustre les composants de la carte à puce

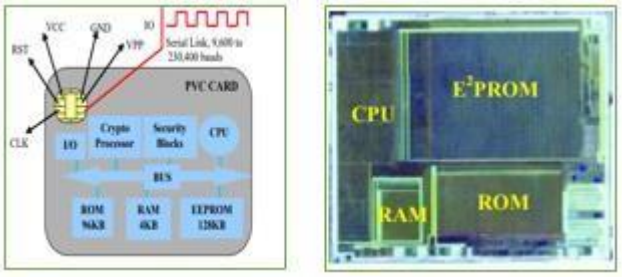
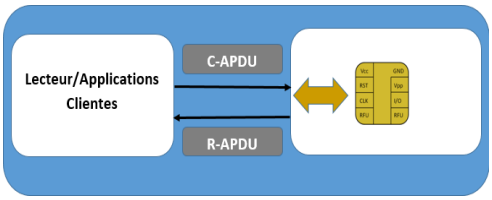


Figure 4- *Vue sur une puce [03]*

1.4.Communication avec l'extérieur (Protocole APDU/TPDU)

La communication entre l'hôte et la carte est semi-duplex et elle se fait à l'aide de paquets appelés APDU (*norme ISO 7816-4*). Ces paquets contiennent soit une commande soit une réponse entre la carte et le lecteur. La carte dans ce dernier joue un rôle passif et attend une commande APDU. Elle exécute l'instruction spécifiée dans la commande et retourne une réponse APDU. Donc, le terminal est toujours l'initiateur de la commande. Le protocole de communication de la carte à puce est montré dans la figure 5. *[05]*

Figure 5- *Le modèle de communication de la carte à puce. [05]*



1.4.1. Commandes APDU

Entête obligatoire				Corps optionnel		
CLAS	INS	P1	P2	Lc	Données	Le
1 Octet	1 Octet	1 Octet	1 Octet	1 Octet	Non définie	Non définie

Tab 2 : *Format des commandes APDU*

- **CLA** : classe ;
- **INS** : code de l'instruction ;
- **P1, P2** : paramètres de l'instruction ;
- **Lc** : nombre d'octet présents dans le champ de données ;
- **Données** : données à envoyer vers la carte ;
- **Le** : nombre d'octet à recevoir de la carte ;

Suite aux données que portent ces commandes, Il existe 5 types de commandes APDU selon qu'il y'a ou non échange de données utiles (voir tableau 2). *[04]*

1.4.2. Les réponses APDU

Après la réception de la commande APDU par la carte, ce dernier répond en envoyant le code instruction INS, suivi de données de longueur variable en terminant par SW1 et SW2 (*0x90 0x00 lorsque la commande s'est déroulés avec succès*). En cas d'échec, seuls les champs de statut SW1 et SW2 seront envoyés au terminal. *[08]*

- **SW1, SW2 :** Status Words (*mots d'état*) donnent l'état de traitement par la carte ;
- **Données :** données reçues de la carte.

Corps optionnel	Partie obligatoire	
Données	**SW1**	**SW2**
Varie	**1 octet**	**1 octet**

***Tab 1.3** : Format des Réponses APDU.*

a) Protocole TPDU (Transmission Protocol Data Unit)

Les données échangées entre le lecteur et la carte avec les commandes et réponses APDU sont référenciées comme TPDU pour Transmission Protocol Data Unit. Comme il a été mentionné précédemment, on trouve principalement deux formats de TPDU. *[09]*

- Le premier consiste à envoyer des données depuis le lecteur vers la carte à puce. Comme l'explique la figure 6.
  - Le lecteur commence par envoyer la classe, l'instruction et ses paramètres.
  - La carte répond en envoyant le code de l'instruction pour indiquer qu'elle a bien reçu la commande.
  - Le lecteur compare le code reçu de la carte avec celui de la commande qu'il a envoyé, s'il est le même, le lecteur envoi l'autre moitié de la commande qui correspond à la donnée d'une taille P3.
  - Après la réception, la carte répond en envoyant SW1 et SW2 afin d'indiquer l'état du déroulement de la commande.
- Le deuxième consiste à envoyer des données depuis la carte à puce vers le lecteur. Comme l'explique la figure 7.
  - Le lecteur commence par envoyer la classe, l'instruction et ses paramètres.
  - La carte répond en envoyant le code de l'instruction pour indiquer qu'elle a bien reçu la commande.
  - Le lecteur compare le code reçu de la carte avec celui de la commande qu'il a envoyé, s'il est le même, le lecteur attend la réception d'une donnée d'une taille P3.
  - Après la soumission de la donnée, la carte envoi SW1 et SW2 afin d'indiquer l'état du déroulement de la commande.

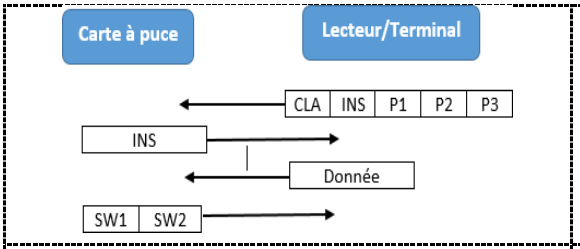


***Figure 6**- Donnée envoyé depuis le lecteur vers la carte à puce. [08]*

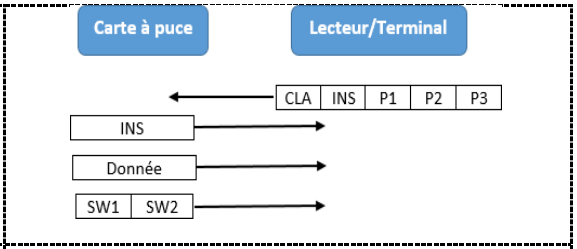


***Figure 7**- Donnée envoyée depuis la carte à puce vers le lecteur. [08]*

b) Identificateur d'application (*AID*)

Dans la technologie des cartes à puce, chaque application possède un identifiant capable de l'identifier parmi les autres applications. Afin de transmettre une commande à une Applet déployée dans une Java Card, il est nécessaire de la sélectionner par l'envoi d'une commande

APDU spécifique. Cette commande précise l'identifiant de l'application (AID), qui est composé de deux parties : [07]

- La première sur 5 octets, le RID (*Registered application provider Identifier*), est assigné par la norme ISO. Celui-ci correspond à l'identifiant propre à l'entreprise.
  Le second, PIX (*Proprietary Identifier eXtension*) est quant à lui composé entre 0 et 11 octets. Chaque entreprise gère l'affectation des PIXs pour leur AID.

Figure 8- *L'identifiant d'application.*

Application Identifier(AID)	
RID (5 octets)	PIX (0-11 octets)

1.5.La technologie JAVACARD

Généralement, les cartes à puces sont caractérisées par des ressources très limitées disponibles pour l'exécution des opérations arithmétique ou encore des applications toutes entières, et ce, en terme de mémoire et de calcule (*CPU*). En outre, l'explosion des réseaux de télécommunication et des transactions électroniques (*financières, administratives, médicales, etc*.) ont augmenté le besoin en applications sans négliger la sécurité, d'où la nécessité de pensé aux solutions légère et fiable en terme de calcule et en terme de sécurité. *[10]*

Il existe différentes plateformes logicielles ouverts permettant d'utiliser les cartes à puce (*Windows for SmartCard, Multos, Basic Card, Java Card, Linux*) *[09]*. Java Card, proposé par SunMicrosystems Inc en 1996, est l'une des plateformes les plus utilisées sur le marché. En 2018,près de six (06) milliards de cartes à puce basées sur Java Card ont été déployés par les titulairesdes licences Java Card, ce que représentent environ 90% des cartes à puce dans le monde. *[11]*

Java Card est un sous-ensemble des technologies Java adapté aux objets à faibles ressources comme les cartes à puce. Il s'agit d'une plateforme portable, sécurisée et multi-applicative qui intègre les avantages du langage Java :

- la programmation orientée objet offerte par Java;
- Une plateforme ouverte qui définit des interfaces de programmation (*APIs*) et un environnement d'exécution standardisé ;
- L'utilisation d'une machine virtuelle permettant la portabilité des applications et une sécurité accrue lors de l'exécution du code ;
- Une plateforme qui encapsule la complexité sous-jacente (*assembleur*) et les détails du système des cartes à puce.

L'appellation d'une carte à puce par 'Java Card' revient donc à toute carte à puce possédant une plate-forme basée sur la technologie Java Card. Elle est dite «*ouverte*», car elle permet de charger et d'exécuter des programmes écrits en Java. Contrairement aux cartes à puce traditionnelles, les programmes exécutés par la carte ne sont pas forcément fournis par l'émetteur de la carte. ***Dans notre travail de recherche on a focalisé les recherches sur la technologie Java Card car cette technologie représente presque 90% des cartes au niveau mondial****.[12]*

La figure 9 illustre le mécanisme de transfert d'une application vers la carte. L'entité exécutable en carte à puce est appelée «*Applet*». Une applet Java Card est un programme Java qui adhère à un ensemble de conventions qui lui permettent d'être exécuté dansl'environnement d'exécution Java Card. Cette Applet respecte la norme ISO 7816, c'est-à-dire qu'elle répond à des requêtes de la même manière qu'elle les reçoit sous la forme de commandesen byte codes appelées des Commandes APDU *[14]*.

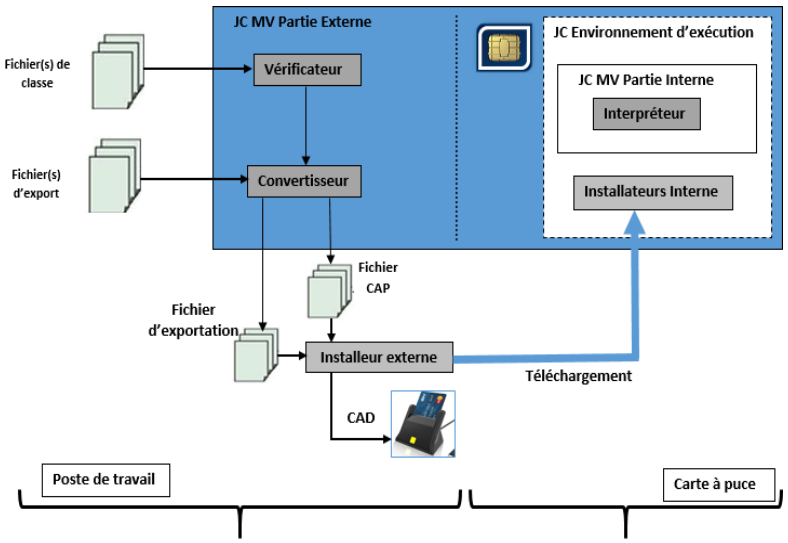


Figure 9- *Mécanisme de transfert des fichiers dans la carte à puce.[45]*

1.6.Mécanismes de sécurité des cartes à puce

La technologie implémentée dans les smart cartes (*Java Card, Cryptographies, Signature électronique, etc.*) a apporté des innovations majeures et extraordinaires, qui ont permet une flexibilité pratique et opérationnelle. En revanche, la sécurité de ces systèmes et technologies est classée comme une préoccupation importante par les concepteurs au niveau mondiale. C'est un sujet qui a constitué un défi majeur, et ce, dû aux contraintes liées aux dispositifs matérielset logiciels spécifiques, notamment la limite des ressources (*Mémoires, espaces, etc.*) et les contraintes de puissance (*Processeur, unités de calcules, etc.*).

En effet, le défi de la sécurité de cette technologie c'est de garantir et d'assurer que l'exécution des applications ne compromette pas l'intégrité de la carte à puce ou la confidentialité des autres données qui y sont stockées sur cette cartes, notamment les données à caractères personnelles, financières ou encore les données techniques qui servent au processus de cryptage ou décryptage. La sécurité des cartes à puce rassemble les mécanismes de la sécurité physique et les mécanismes intégrés à la plateforme java card elle-même ainsi que la sécurité offerte par le langage de programmation Java spécifique à cette technologie et approprié au développement des applications pour cartes à puce. Ceci est résumé dans la figure 10.

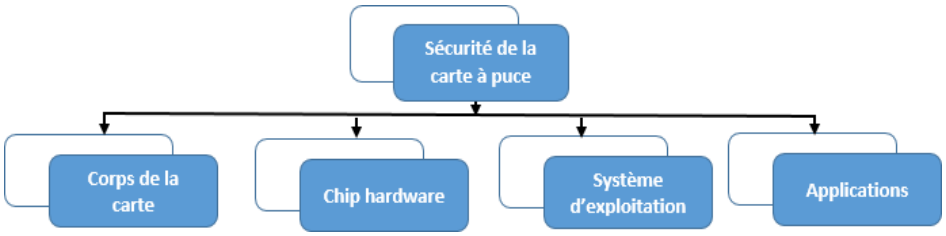


Figure 10- *Classification des composantes de sécurité des cartes à puce.[10]*

1.6.1. Sécurité matérielle

La sécurité matérielle des cartes à puce est un des éléments clefs et primordiale de la sécurité des informations sensibles et confidentielles qu'ils les stockent ou qu'ils les traitent. Dans ce contexte, les fabricants de ces cartes à puce sont très conscients des enjeux et les menaces qui ciblent ces moyens électroniques, et ils prêtent une attention particulière à la sécurité de leur produit. La puce elle-même est fabriquée et placée dans une carte plastique (*construite de plusieurs couches*) de façon à ce qu'il soit très difficile voire même impossible à quiconque d'accéder à ces composantes internes. *[10]*

1.6.2. Sécurité logicielle

Les cartes à puce se présentent comme un petit ordinateur qui contient un système d'exploitation, souvent appelé l'environnement d'exécution, des unités d'entrées sorties, des zones de stockages (*mémoires*), ainsi des applications qui servent à répondre à des requêtes de la même manière qu'elle les reçoit sous la forme de commandes depuis l'extérieur (*lecteurs*). Ces cartes doivent donc être sécurisées aussi bien d'un point de vue physique ou matériel que d'un point de vue logiciel. *[10]*

1.6.3. La Cryptologie et la carte à puce

La carte à puce est un moyen utilisé pour garantir un niveau très élevé d'authentification et d'identification dans plusieurs domaines. Ces processus sont basés sur la cryptographie pour effectuer des opérations comme le chiffrement, le déchiffrement, l'échange des clés et la vérification des signatures numériques. Ces techniques de chiffrement sont utilisées aussi lors du stockage ou de l'échange d'informations sensibles avec l'extérieur. Donc, la cryptographie offre pour les cartes à puce les objectifs de sécurité, notamment : La confidentialité, l'intégrité, l'authentification et la non-répudiation *[15].* Actuellement, plusieurs algorithmes cryptographiques symétriques ou asymétriques (figure 11) sont utilisés dans le domaine des cartes à puce, et ce, pour permettent de renforcer les mécanismes de sécurité existes. Parmi ces algorithmes, on cite : DES, AES, RSA et ECC. Ces algorithmes sont implémentés en logiciel à savoir des applications de chiffrement supportées par les cartesà puce, ou bien en matériel à savoir les cryptos-processeur, qui est utilisé pour faire les grandscalculs liés aux algorithmes cryptographiques. *[14]*

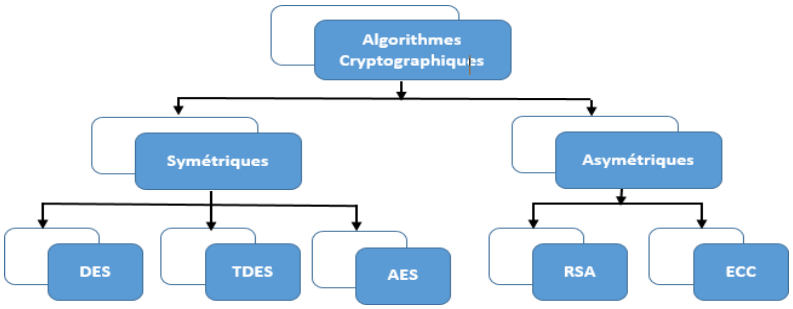


Figure 11- *Algorithmes cryptographiques utilisés dans l'environnement de carte à puce. [04]*

1.6.4. Crypto processeur des cartes à puce

L'utilisation des algorithmes de cryptographie avec la manipulation de grands nombres comme les algorithmes à clefs symétriques ou asymétrique, a constitué une source de consommation de puissance de calcul. C'est pourquoi la technologie des cartes à puce a utilisé, de plus en plus, un second crypto processeur sécurisé adjoint au microprocesseur traditionnel de la carte à puce qui accélère les calculs sur les grands nombres lié à la cryptographie.[16]

Donc, pour des exigences spécifiques liées à la cryptographie sur cartes à puce, les fabricants de microcontrôleurs de cartes à puce offrent une large gamme de fonctions et composantes supplémentaires sous la forme de matériel sur puce, notamment l'adoption de ces crypto processeur à côté des processeurs classiques et des Générateur de nombres aléatoires (*RNG*). *[17]*

1.6.5. Sécurité de la plateforme Java Card :

Elle est assurée par un ensemble de composantes et mécanismes, chacun d'entre eux assure une partie de la sécurité tout en étant complémentaire avec les autres pour offrir une sécurité maximale à la carte. Les principaux éléments de sécurité intégrés disponible dans Java Card se résument dans ce qui suit :

- **Le vérifieur du sous-ensemble des fichiers. Class**

Comme la technologie Java Card est un sous-ensemble du langage Java, le convertisseur doit vérifier également que les fichiers .class n'utilisent que les caractéristiques de ce sous ensemble. Cette étape est appelée vérification du sous-ensemble (*subset-checking*). Durant cette étape le convertisseur vérifie la validité des règles : types non supportés (*long, double et float*), threads, tableaux multidimensionnels, etc. *[18]*

- **Le vérifieur de bytecode**

Ce vérifieur est considéré comme une composante cruciale, et ce, en vue qu'il est le processus offensif de la sécurité de Java Card, du fait qu'il intervient avant le chargement des applets sur la carte. Il a pour objectif de s'assurer que le code à charger peut être exécuté sans risques par la machine virtuelle et ne peut pas outrepasser les mécanismes de sécurité de haut niveau. *[03][05]*

- **Processus de chargement des applets**

L'installeur c'est le responsable du téléchargement sécurisé des applets sur la carte après qu'elle soit délivrée au titulaire. Cet installeur est utilisé pour des buts de chargement des applets de sur une des mémoires qui existes sur la carte à puce et dédiée a logé ces aplets.

Global Platform définit une entité, située sur la carte, appelée Gestionnaire de carte (*Card Manager*) qui est capable d'effectuer un chargement plus sophistiqué. Généralement, le gestionnaire de carte met en oeuvre un protocole de canal sécurisé qui fournit des services cryptographiques comme le chiffrement et l'authentification pour garantir l'intégrité et la confidentialité de l'applet lors du transfert et pour authentifier le serveur depuis lequel l'applet est chargée. *[19]*

- **Pare-feu et contextes**

Une Java Card peut supporter plusieurs applets qui peuvent provenir de différents vendeurs, de sorte qu'il n'est pas sécurisé si les objets d'une applet sont accessibles à une autre applet. Le pare-feu des Java Cards est mis en oeuvre dans le JCRE, Il a pour mission d'isoler les applets et les différents objets créés au sein d'espaces protégés appelés contextes. *[20]*

- **Atomicité des transactions**

Java Card introduit un mécanisme de transaction qui garantit son atomicité. Une opération de validation à la fin de la transaction confirme l'achèvement des opérations précédentes. Si la transaction est annulée (par une coupure de courant par exemple), le mécanisme permet de s'assurer que toutes les opérations précédentes au sein de la transaction soient inversées. De cette façon, il est possible de maintenir la cohérence interne des données connexes. *[21]*

1.7.Utilisation des cartes à puce en Algérie

Comme il a été mentionné précédemment, l'Algérie a déployé ces dernières années, plusieurs solutions et services électroniques au profit des citoyens. Ces solutions sont conçues àla base de l''utilisation des cartes à puces, tels que le paiement électronique dans le domaine bancaire et monétaire, l'utilisation de la carte nationale d'identité biométrique et le passeport biométrique par les services relevant du ministère de l'intérieure ainsi que l'utilisation de la carte d'assurance Chiffa par la caisse nationale des assurances sociales.

Au niveau mondial, l'utilisation des cartes à puce dans le domaine bancaire et financière a connu plusieurs phénomènes de fraudes et d'attaques qui ciblent ces moyens électroniques. En effet l'objectif de ces attaques est pour accéder aux informations confidentielles existes sur les cartes (Puce, Bande, Donnée Embossées, etc.) et de faire des transactions électroniques frauduleuses et illicites. Ces techniques d'attaques et de fraudes contre ces cartes à puce a connu ces dernières années une explosion catastrophique au niveau mondial *[01]*.

Dans ce travail on s'intéresse beaucoup plus aux transactions électroniques financières réalisées par les cartes à puce ainsi que certaines typologies de fraudes et d'attaques menaçants la sécurité de ces dernières.

L'utilisation des cartes à puce dans le domaine bancaire en Algérie se résume dans les cartes de type CIB et EDAHABIA d'Algérie poste qui servent à faire plusieurs transactions financières électroniques comme les achats, les paiements en ligne, le paiement de proximité, ainsi des transactions financières via les distributeurs ou guichets automatiques de billet, de carburant, autoroute, etc. (figure 12)

Les transactions électroniques réalisées via ces moyens ont été renforcées par plusieurs standards et normes de sécurités internationales comme le standard EMV (*Europay Mastercard Visa*), qui est un standard international lié à la sécurité des cartes de paiement dotées des puces électroniques. D'autres protocoles ont étés déployés pour garantir la sécurité des données des cartes lors d'une transaction en ligne et d'assurer aussi la bonne authentification du porteur de la carte lors de ces transactions, à savoir, le protocole 3DS et 3DS2 ou encore EMV-3DS. *[22]*

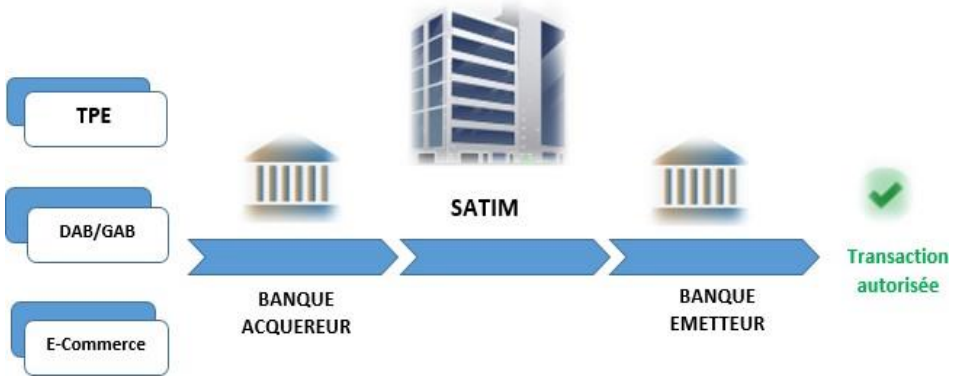


Figure 12- *Ecosystème du réseau monétaire en Algérie [22]*

1.8.Standard EMV (Europay Mastercard Visa)

1.8.1. Description du standard EMV

EMV est un standard international de sécurité des cartes de paiement dotées des puces électroniques. C'est un ensemble de spécifications qui définit est déterminé la sécurité de ces cartes à puce. EMVco est l'organisme qui s'occupe de ce standard est qui regroupe : MasterCard, Visa, JCB et American Express. Des améliorations très importants de ces spécifications liées à la sécurité ont été adoptés à savoir l'interopérabilité internationale, la vérification de la clé personnelle (*PIN*) par la puce, opérations de chiffrement et déchiffrement ainsi que le multi-applicatif. Plusieurs applications peuvent alors cohabiter sur la même carte à puce. Ces mesures ont permis la diminution de la fraude qui a touché les anciennes cartes de paiement basées initialement sur les pistes ou bandes magnétiques. Le standard EMV couvre la transaction à partir de l´émetteur jusqu'au terminal en passant par l'acquéreur. *[23]*

Le standard EMV permet l'utilisation d'un code PIN et des algorithmes cryptographiques tels que DES, Triple-DES, RSA et SHA afin de permettre à la carte et au terminal de s'authentifier mutuellement. Ces opérations cryptographiques jouent un rôle primordial dans le processus de prévention contre la fraude. Ledit standard s'appuie sur la norme ISO/IEC 7816 pour les échanges entre le lecteur et la carte, notamment 7816-3 et 7816-4, précédemment citées, et qui normalisent les protocoles de transmission ainsi que la définitiondu format des échanges au niveau applicatif. Les principales commandes utilisées par ce standard, on cite :

- *application block.* - *application unblock.*	- *gênerait application cryptogram.* - *get data (7816-4).*	- *PIN change / unblock.* - *read record (7816-4).*
- *card block.* - *external authenticate (7816-4).*	- *get processing options.* - *internal authenticate (7816-4).*	- *select (7816-4).* - *verify (7816-4).*

La spécification de l'EMV garantie que quel que soit le fabricant de la carte à puce, celle-ci doit être lue par n'importe quel distributeur dans le monde. Pour garantir cette interopérabilité, la normalisation concerne au moins trois (3) points liés généralement aux paramètres physiques : taille de la carte, position de la puce et ses contacts, des paramètres électriques : tension d'alimentation, niveaux électriques utilisés et finalement des paramètres logiciels qui définissent le mode de dialogue avec la carte (*commandes*).

De point de vue de la sécurité, le standard EMV vient pour mieux combattre les typologies de fraudes qui ont ciblé les anciennes cartes à puce magnétique dont les processus d'authentification se base principalement sur des données statiques existes sur ses pistes magnétiques. Généralement les principaux scénarii de fraudes à la carte de paiement se résument en : La contrefaçon, La duplication de carte (*clonage*), utilisation de cartes perdues ou volées, utilisation illégitime des données de carte dans un contexte de carte non présente. *[24]*

1.8.2. Principales étapes du traitement d'une transaction EMV

Basé sur la norme iso/7816, le système de fichier d'une carte EMV comporte des répertoires DDF (*Directory Definition File*) et des répertoires ADF (*Application Definition File*) dédiés à des applications particulières, qui contiennent des fichiers AEF (*Application Elementary File*). Les éléments d'information mis en œuvre par les applications sont partiellement décrits par la norme EMV, cette dernière fournit donc une architecture de référence, mais permet un certain degré de liberté aux applications de paiements.

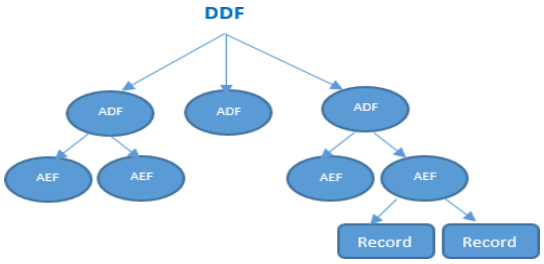


Figure 13*- Le système de fichier d'une carte EMV [03]*

A la racine, il existe un répertoire DDF particulier et considéré comme l'index de tous les sous répertoires. Pour les systèmes de paiement, le PSD (*pour Payment System Director*) nommé «1PAY.SYS.DDF01», permet d'obtenir le mode d'emploi de la carte EMV, et l'accès aux autres sous répertoires contenus la DATA, c.-à-d les données élémentaires de la carte. Le répertoire racine est activé grâce à la commande SELECT. L'accès aux données élémentaires existes au niveau des record se fait par la commande READ RECORD. *[03]*

Afin de communiquer avec une carte, la norme EMV basée sur le standards ISO/IEC 7816 définit les commandes APDU à envoyer pour établir la communication entre le terminal et la carte. Deux protocoles existent : T=0 qui utilise une transmission de donnée par caractèreet T=1 qui utilise une transmission de donnée par blocs structurés. Le début et le déclanchementd'une connexion entre la carte et le terminal se fait par une réponse à un signal électrique émispar le terminal sur le contact RST du micromodule. La carte renvoie une réponse normalisée

nommée ATR (*Answer To Reset*) et les informations nécessaires au terminal pour établir la communication comme la vitesse et protocoles d'échanges. Après, pour permettre au terminal de déclencher certaine action et transactions, ce dernier utilisera plusieurs commandes APDU qui forment une session de communication et d'échange de données. *[13]*

Toutes les étapes d'une transaction EMV se traduisent en un certain nombre de commandes APDU qui peut être envoyées par le terminal et exécutées par la carte, et ce, pour garantir la sécurité et l'atomicité de cette transaction. *[13]*

- *Power On* ; le terminal carte fournit du courant à la carte et commence le processus d'initialisation. La carte répond par un Answer To Reset ;
- *Select Application* ; le terminal sélectionne une application ;
- *Get Processing Options* ; le terminal demande des informations à la carte sur son contenu ainsi que des données fonctionnelles pour le reste de la transaction ;
- *Read Record* ; le terminal va lire aux endroits appropriés dans le système de fichier des informations supplémentaires sur l'application sélectionnée ;
- *Get Challenge* ; le terminal demande à la carte un nombre aléatoire à la carte pour diversifier ou chiffrer une opération future ;
- *Internal authenticate ;* authentifie l'application pour que le terminal puisse accéder à des données sensibles de la carte ;
- *Verify Pin* ; le porteur est authentifié auprès de la carte ;
- *Generate Application Cryptogram ;* un cryptogramme d'autorisation est généré et la transaction finalisée.

1.9.Le système d'authentification 3Dsecure (*3DS*)

1.9.1. Description du système d'authentification 3Dsecure

Le système d'authentification 3D Secure est un système de sécurisation des paiements en ligne, créé par les émetteurs internationaux Visa et MasterCard. Ce système a été mis place pour s'assurer la bonne authentification du porteur de la carte, et ce, pour s'assurer que c'est bien celui-ci qui effectue le paiement sur internet et n'est pas une personne malveillante. En plus les avantages garantis par le standard EMV, ce protocole sert à compléter les principaux objectifs de de sécurité à savoir, la confidentialité, l'intégrité et l'authentification. Pour accomplir une transaction de paiement sécurisée par le protocole 3Dsecure, il faut achever les étapes suivantes :

<u>Étape 1</u> : saisie des informations bancaires

Comme pour toute transaction sur un site marchand, dans l'espace de règlement de la commande, le porteur de la carte doit saisir les informations suivantes :

- Le numéro de la carte bancaire,
- La date de validité de celle-ci,
- Le cryptogramme visuel (*3 chiffres figurant au dos de la carte*).
- Informations personnelles.

<u>Étape 2</u> : l'authentification de la transaction

Après l'acheminement de l'étape précédente, une nouvelle fenêtre s'ouvre alors, invitant le porteur de la carte à confirmer son identité, avec le procédé d'authentification mis en place par celui-ci. Selon la procédure la plus usuelle, le système de paiement mise en place envoie un code à usage unique OTP (*One Tome Password*) par SMS. Cette information sera saisie pour confirmer l'authentification par la banque, le paiement est autorisé et la transaction est finalisée.

Les systèmes du paiement en ligne en Algérie mises en place par la Société d'Automatisation des Transactions Interbancaires et de Monétique SATIM et Algérie Poste reposent sur les procédures d'authentification 3D Secure. Elles peuvent créer une couche supplémentaire de protection contre la fraude et s'assurer d'accepter uniquement des paiements par carte de clients légitimes.

1.9.2. Limites du 3DS

Pour les transactions authentifiées par le protocole 3D Secure (**Même pour les systèmes de paiement mises en place en Algérie**), la confirmation de celle-ci par un SMS reçu sur son téléphone portable, indique seulement que ce message affiché au propriétaire de la ligne de téléphone a été recopié sur une page web du site marchand. Mais ne suppose aucunement de l'engagement du propriétaire de la ligne car ce dernier n'a pas été authentifié (*cas de l'utilisation du portable par un tiers*). Autrement dit, aucune preuve matérielle ne permet de s'assurer de son engagement dans la transaction. *[25]*

Les systèmes de paiement au niveau mondial, notamment en Europe, ont amélioré leurs mécanismes en se basant sur l'authentification à plusieurs facteurs. Ce dernier consiste en un processus d'authentification pour les transactions en ligne qui s'appuie sur au moins deux facteurs. Ces facteurs incluent soit quelque chose que l'on possède (*comme un mot de passe unique à six chiffres*), quelque chose que l'on connaît (*comme un mot de passe, des identifiants ou les réponses à des questions de sécurité*) ou quelque chose que l'on « est » (*comme des identifiants biométriques*). Par exemple dans un processus type de code de vérification à deux étapes, un utilisateur entre ses identifiants de connexion habituels. Par la suite, le système valide la transaction par l'envoie d'un code de vérification ou une clé de sécurité par message texte au numéro de téléphone de l'utilisateur plus un autre facteur qui doit être entré par l'utilisateur comme la biométrie. *[25]*

1.10. Le système d'authentification 3DS-2

1.10.1. Description du système d'authentification 3DS-2

A causes des limitations constatées sur la 3DS, les systèmes de paiement dans l'union européenne ont commencé à partir le mois de janvier 2021, a remplacé cette authentification par une mise à jour de la 3DS c'est la 3DS2 ou encore EMV-3DS. Le nouveau mode d'authentification avec ce protocole est fondé sur l'authentification basée sur l'analyse des risques. Suite au niveau de risque de la transaction, le système prend la décision pour la poursuite de cette transaction ou le passage à d'autres étapes d'authentification, en demandant des données d'authentification supplémentaire.

En effet par l'utilisation du 3DS2, les transactions sont filtrées pour des éléments qui les placeraient dans différentes catégories de risque. Par exemple si le système détecte qu'une nouvelle carte est utilisée par un utilisateur sans historique de transaction, le risque sera probablement jugé élevé d'où une nécessité de processus d'authentification en demandant des informations supplémentaires. Parmi ces éléments fondés sur les risques on cite ce qui suit : *[26][27]*

- La valeur de la transaction.
- Client nouveau ou existant.
- Historique transactionnel.
- Antécédents comportementaux.
- Informations sur l'appareil, etc.

Étape 1 : L'utilisateur saisie des informations bancaires lorsqu'il est prêt à effectuer un paiement.

Étape 2 : Les détails de la transaction ainsi qu'une demande de vérification seront envoyé par le système de paiement à la banque émettrice.

Étape 3 : L'émetteur doit déterminer la possibilité d'effectuer une authentification sans informations supplémentaire saisie par l'utilisateur.

Étape 4 : Si l'émetteur détermine que la transaction est à faible risque et qu'une authentification sans friction est possible, il authentifie l'opération. Par contre si cet émetteur détermine que la transaction est à haut risque il passe à la demande des informations d'authentification supplémentaires de l'utilisateur. (Authentification forte)

Étape 5 : A la fin l'utilisateur reçoit la confirmation du paiement réussi sur le site du commerçant.

1.10.2. Limites du 3DS2 :

Généralement la 3DS2 est basé sur la collecte de données complexes pour l'analyse des risques *(La valeur de la transaction, historique transactionnel, client nouveau ou existant, antécédents comportementaux, informations physiques sur l'appareil (ordinateur, mobile, tablette, etc.), information de géolocalisation, etc.),* et ce, dans le but d'identifier les risques et les activités frauduleuses par l'utilisation des données d'une carte d'une manière illicite. Ce processus permet la sauvegarde d'immense information sur les porteurs de cartes par les E-marchant pour réduire les risques liés aux paiements non authentifiés. Des travaux de recherches très récents ont conclu que ces manouvres présentent des failles de sécurité fondamentales qui peuvent permettre au commerçant de se faire passer pour le titulaire de la carte, lorsqu'il est utilisé dans une configuration où le titulaire de la carte utilise une application native pour accéder au site du commerçant. Donc les risques d'utilisation de ces données par les E-marchants malveillants restent très probables pour passer des transactions frauduleuses. [48]

Après avoir présenté les concepts de base liés à la carte à puce et sa sécurisation, nous allons présenter dans la section suivante les différents types que peuvent subir ce genre de cartes.

Section 2 : Les attaques contre les cartes à puce (Cas des cartes bancaires)

Généralement, une attaque consiste à exploiter et à utiliser les caractéristiques ou les particularités de la cible à attaquer, notamment ceux qui contient de faiblesse ou de vulnérabilités constatables. L'objectif c'est de tenter de contourner les mécanismes de sécurité matériels ou logiciels qui lui sont intégrés, et accéder aux informations et aux secrets contenus dans la cible. Pour les cartes à puce les attaquants cherchent à exploiter les failles qui existent aux niveaux de ces cartes que ce soit au niveau physique, système et applications ou encore à tirer profit de la vulnérabilité du porteur. *[10]*

Il existe plusieurs approches pour la classification systématique des attaques sur les cartes à puce(figure 14). En revanche, la classification citée dans notre travail peut être divisées en trois types différents d'attaques, et ce, suite aux modes opératoires et aux parties ciblées par las attaquants :les attaques au niveau physique, les attaques au niveau logique, et les attaques de types ingénierie sociale qui ciblent les données embossées sur cartes. Les deux premiers sont considérés comme des attaques internes à la carte, par contre le troisième type est considéré comme externe à la carte. *[29]*

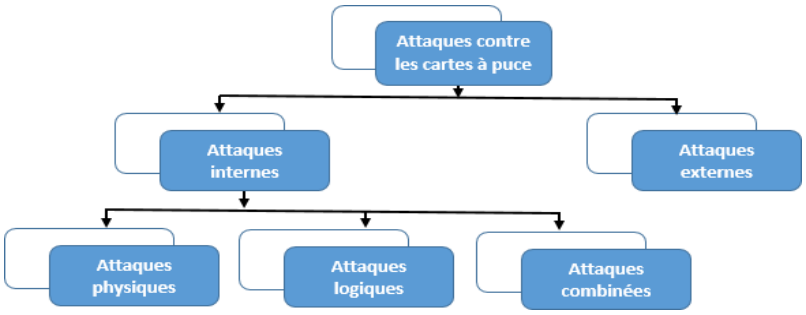


Figure 14- *Classification des attaques contre les cartes à puce.*

Depuis son existence, la carte à puce et/ou à bande magnétique a été la cible de plusieurs mécanismes et typologies d'attaques physiques et logiques, et ce, pour accéder aux données confidentielles (*code PIN, clés secrètes de cryptographie, N° de carte, Date de validité, Etc.)* et de faire des transactions électroniques frauduleuses et illicites (*Transactions financières, administratives, d'authentifications, d'identification, Etc.*) *[30]*. Ces techniques de fraudes et d'attaques ont ciblé les endroits où il existe ces données confidentielles, à savoir la bande magnétique, la puce électronique, ou encore les données embossées sur ces types de cartes l'ors d'une transaction (*En ligne ou en proximité*). ***<u>Ci-dessous, nous mentionnons la classification de ces attaques ainsi certains types de ces attaques contre les cartes à puce (Exemples)</u> :***

Figure 15- *Les cibles d'attaques (Bande magnétique, puce électronique, données embossées)*

2.1.Attaques internes

2.1.1.Attaque contre la bande magnétique

En effet l'objectif du passage de la bande magnétique à la puce électronique c'était de trouver des nouvelles technologies pour faire combiner et cohabiter un moyen de stockage d'information et au même temps un mécanisme de calcul puissant permettant une manipulation

contrôlée et sécurisée de données personnelles et secrètes. Avec l'utilisation de la bande magnétique, considéré faible en termes de sécurité, plusieurs techniques ont été exploité afin accéder aux informations confidentielles existes sur cette bande.

Prenant l'exemple de la bande magnétique qui existe sur certain carte à puce, notamment les cartes bancaires. Cette bande contient en réalité des informations confidentielles du propriétaire de la carte et données bancaires, et qui existent au même temps sur la puce électronique. Ces données sont lues par le terminal de paiement et/ou certain distributeur, (suivant le système utilisé) pour effectuer des transactions financières. Cette bande magnétique a constitué, depuis son utilisation, le point faible de ce type de carte bancaire, du fait que les données statiques qui sont stockées sur cette bande, sont facilement accessibles etdupliables. *[30]*

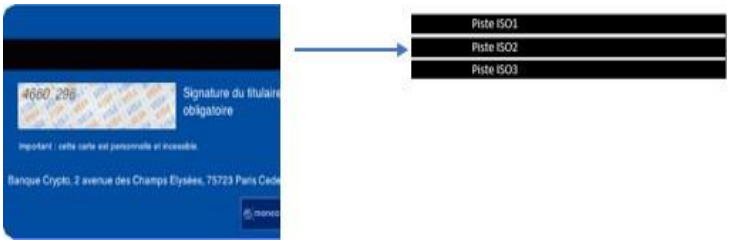


Figure 16 : la bande magnétique. [30]

- **Le skimming des bandes**

L'attaque d'une carte bancaire par l'exploitation de la bande magnétique, est une techniquesimple qui a été largement exploitée au niveau mondial, il s'agit du «Skimming». L'attaquant a besoin de deux types d'informations : les données stockées sur la bande magnétique (*N° de la carte, date d'expiration, données du propriétaire et autres*) et le code personnel à quatre chiffres. Ces informations peuvent être récupérées par l'exploitation les endroits où ces cartes sont utilisées, tels que des distributeurs automatiques de billets, les distributeurs automatiques de carburants, les terminaux de paiements, etc. *[30]*

- Les données de la bande magnétique sont organisées en trois pistes, répondent aux spécifications de la norme iso (piste ISO1, piste ISO2, piste ISO3). Ces données peuvent être lues ou modifiées avec une tête de lecture/écriture magnétique. Avec le remplacement du lecteur decarte par un faut dispositif (*dispositif pirate*) : la fente censée accueillir la carte bancaire n'est plus celle du distributeur, mais celle conçue par un pirate. Elle est reliée à un équipement qui enregistre les données de la bande magnétique, autrement dite un signal analogique. Ces données seront utilisées par les fraudeurs.

Figure 17- *Le dispositif du skimmer (mini-caméra et faut lecteur). [31]*

- Avec un dispositif de surveillance caché pour obtenir le code à quatre (4) chiffres : il s'agit généralement d'une micro-caméra cachée dans le plafonnier du distributeur automatique. Certains skimmers utilisent également de faux claviers numériques, posés par-dessus le clavier original. Ces claviers transmettent à distance les codes saisis par les utilisateurs.

  Il est également possible de trouver un lecteur/enregistreur de fichier audio installé au niveau des dispositifs de piratage et d'attaque. Dans ce cas, le signal analogique provenant de la tête de lecture est directement enregistré sous la forme d'un fichier audio qui peut être lu avec des logiciels ou un éditeur audio spécialisé, et ce, pour observer les variations et décoder les données binaires. Des outils permettent d'extraire de ces fichiers audio les données binaires des pistesmagnétiques enregistrées en mesurant les espaces entre les variations.

- IL est également possible l'utilisation d'un lecteur/enregistreur de fichier audio au niveau des dispositifs de piratage et d'attaque. Dans ce cas, le signal analogique provenant de la tête de lecture est directement enregistré sous la forme d'unfichier audio qui peut être lu avec deslogiciels ou un éditeur audio spécialisé, etce, pour observer les variations et décoderles données binaires. Des outils permettent d'extraire de ces fichiers audio les données binaires des pistes magnétiques enregistrées en mesurant les espaces entre les variations. (figure 18)

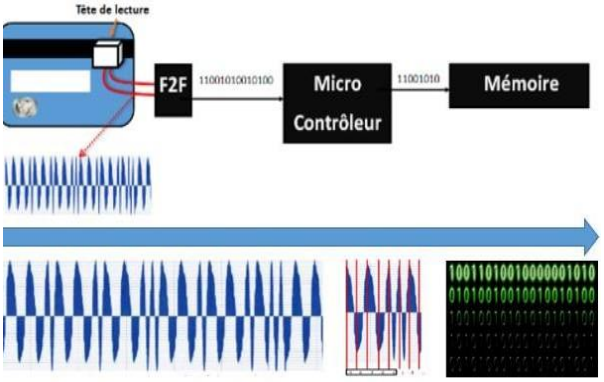


Figure 18- *l'interprétation des données de la piste.* ***[30]***

En pratique, c'est la piste ISO2 qui est utilisée pour le traitement des opérations bancaires et commerciales (*Paiement, Retrait, Etc*.). La suite numérique qui est inscrite sur la piste ISO2 des cartes de paiement et de retrait a une structure particulière qui se présente ainsi :

4552629017174285=0804101127070118000 0
{ *1* } { *2*}{*3*}{ *4* }

1- Numéro de la carte ; les six premiers chiffres de la carte (*Bank identification Number*).
2- Date de validité de la carte.
3- Code service de la carte (*101 internationale à piste/201 internationale àpuce*).
4- Données de sécurité inexploitables. (Cecas)

Une fois les données de la carte obtenues, mes skimmers les copient sur des cartes vierges appelées White Cards, qui porte des bandes magnétiques, et ce, par l'utilisation des moyens et logiciel spécifiques. Ces clones de cartes bancaires sont ensuite envoyés et utilisées dans des pays étrangers dans lesquels les transactionsde paiements et/ou de retrait par carte ne sollicitent que la bande magnétique.

2.1.2.Attaque contre la puce électronique

La sécurité d'une carte à puce peut être contournée de plusieurs manières : soit en prenant le matériel en défaut (*Physiquement*), soit en prenant l'applicatif ou le système en défaut (*Logiquement*). Ainsi nous pouvons classer les attaques connues contre les cartes à puce selon l'arborescence de la figure 20. *[32]*

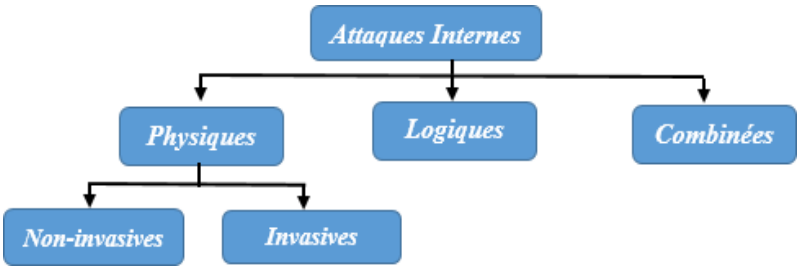


Figure 20- *Classification des attaques contre la puce. [07]*

2.1.2.1.Attaques physiques

Ces attaques touchent la partie matérielle ou physique de la carte à puce, notamment les composantes clés du microcontrôleur tel que les zones mémoires où sont stockées des données à caractère confidentielles. Ces types d'attaques consistent en une analyse et/ou modification des circuits électroniques de la carte, souvent contenant les implémentations des algorithmes cryptographiques, afin d'obtenir des informations sensibles de cette dernière. Cependant, ces types d'attaques nécessitent des ressources importantes comme des outils coûteux et des équipements spéciaux pour attaquer physiquement la carte. Ils peuvent être : invasives ou non invasives. ***Dans ce qui suit on va expliquer quelques techniques les plus exploitées pour passer ces types d'attaques*** : *[32]*

2.1.2.1.1 Attaques invasives

Ces attaques consistent en un accès physique aux composantes internes de la puce afin de réaliser des observations et analyses directes ou encore des modifications de la structure des circuits. Il s'agit d'attaques extrêmement performantes qui touche la partie physique, mais ils présentent cependant l'inconvénient majeur d'être destructive, c'est à dire une fois l'attaque effectuée, la carte devient inutilisable *[32]*. Le but étant de récupérer un ensemble d'informations dela carte en se basant sur une cartographie des circuits. L'attaquant tente de déduire lesalgorithmes utilisés, leurs implémentations, les systèmes de sécurité mis en place et lesinformations contenus dans la puce à partir de l'analyse ou la modification des circuits intégrésdans la carte. Pour arriver à cela, il faut une étape d'isolation physique ou chimique des circuitsde la carte (d'où l'aspect invasif). Plusieurs méthodes et techniques sont exploitées pour réaliserde telles attaques, on cite ce qui suit : *[33]*

- **Produits chimiques et gravure à l'eau forte (*Etching*)** : La technique de l'Etching permet de décapsuler les circuits et isoler les différentes couches. Ce qui permet d'obtenir tous les blocs fonctionnels d'un circuit afin qu'il soit accessible pour analyse et exploitation utile. Alors que les produits chimiques permettent de dissoudre la couche de résine qui fixe les circuits. Après ce processus, la puce est accessible pour une analyse optique ou électrique.

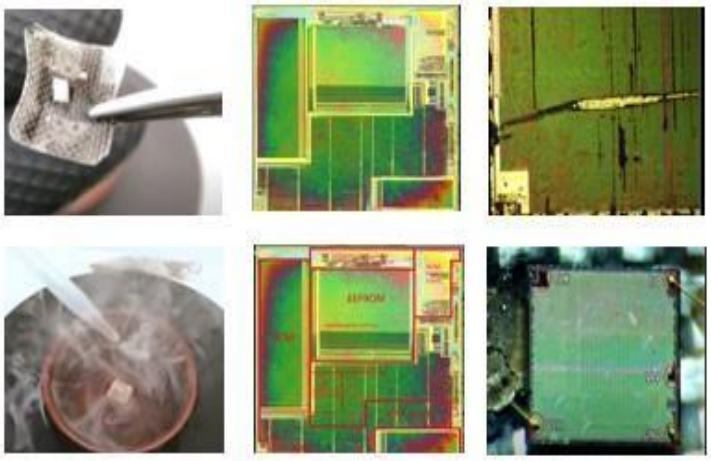


Figure 21 *La décapsulation des circuits et l'isolation des différentes couches de la puce.*

- **Scanning Electron Microscopes (*SEM*)** : Ce type d'attaque consiste en l'utilisation des microscopes pour effectuer de l'analyse optique et du reverse-engineering afin de reconstruire des circuits complets ou le code source d'un programme à partir des bits de la ROM. Ils (*SEMs*) permettent aussi d'observer des circuits durant une exécution d'une commande ou d'un programme. L'analyse des circuits par l'utilisation les techniques du SEM révèle les sections de code qui sont actives et même les valeurs des cellules de la mémoire peuvent être déterminées. En autres, on peut voir les valeurs de tension élevées et faibles (équivalent à 0 ou 1) sur les fils de puce (*Voltage Contrast*). *[32]*

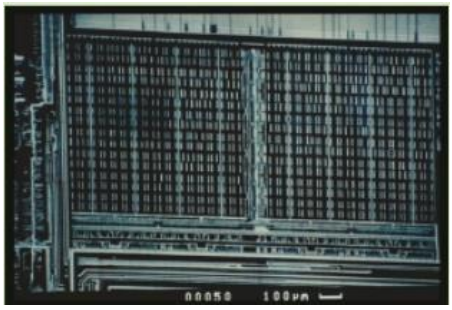


Figure 22- *RAM Voltage Contrast*

- **Le Probing** : Cette technique permet de positionner des micro-sondes, une sorte d'écoute, arbitrairement sur les fils d'un circuit isolé, notamment sur les bus de transmission de données entre les différentes composantes du microcontrôleur. Cette technique permet de créer de nouveaux canaux vers l'extérieur de la carte, et d'obtenir d'information transmis à l'intérieur de la carte. Si le bus de données peut être localisé, les sondes permettront de capter tous les échanges de données entre le CPU et les mémoires. Avec une analyse poussée, ceci permet d'obtenir le code du programme en exécution ainsi que les clés incluses dans les programmes. Il y a aussi une possibilité de modifier les macro-instructions et donc de détourner l'exécution du CPU. *[33]*
- **FIB (*Focused Ion Beam*) :** Cette technique permet de modifier les circuits de la carte en rajoutant ou éliminant des pistes conductrices sur la puce électronique à l'aide de rayons d'ions. Cette technique est utilisée pour atteindre plusieurs objectifs par les attaquants à savoir : la reconnexion de circuits séparés, l'envoi de signaux internes cachés vers l'extérieur et d'avoir de l'information, élargissement des pistes fines et fragiles pour déposer des sondes, etc.

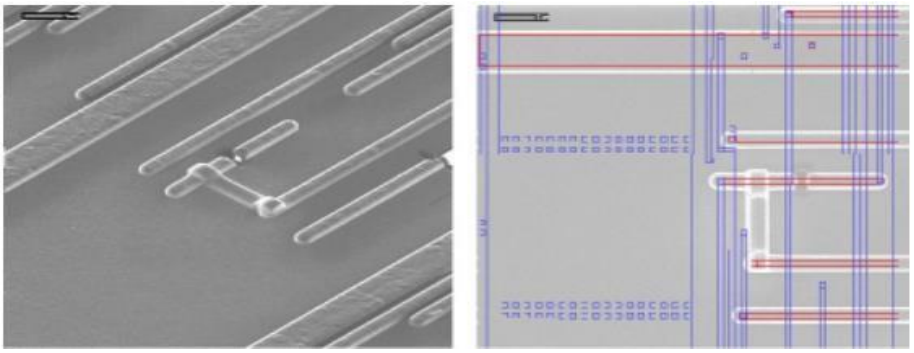


Figure 23- *un circuit modifié au FIB et sa caractéristique [8]*

2.1.2.1.2 Attaques non-invasives

Contrairement aux attaques invasives, Ces types d'attaques ne nécessitent pas de détruire la carte à puce cible pendant l'exécution de l'attaque. Ce type d'attaque consiste en l'analyse d'informations provenant de la carte et extraire les informations clés sans affecter son intégrité et son fonctionnement. Parmi les attaques non invasives, nous distinguons les attaques passives et les attaques actives. Les attaques passives qui ne nécessitent aucune interaction physique avec le matériel attaqué,mais basées uniquement sur ***l'observation*** de l'environnement de la carte et consistent en l'observation des signaux et/ou des émissions électromagnétiques. L'observation dans ce type d'attaque consiste essentiellement à détecter et exploiter des canaux cachés pouvant laisser passer de l'information sur l'état interne de la carte, et en particulier, pour acquérir de l'information sur les secrets contenus dans la carte. [33] Généralement c'est l'exploitation descanaux cachés temporels ou l'analyse du signal de tension d'alimentation : (*Observation- Analyse et exploitation*). Parmi les techniques qui ont été utilisées pour cibler les cartes à puce, on cite:

- **Attaques de type SPA pour Simple Power Attack :** Cette technique d'attaque consiste en une observation directe de la consommation électrique par la carte à puce durant des opérations de cryptages ou d'autres opérations sensibles de sécurité. Ladite technique permettre dans plusieurs cas à révéler des informations aussi bien sur l'opération en cours d'exécution que sur les données traitées. Dans des cas pareils, si un attaquant connaît l'algorithme cryptographique utilisé, il peut facilement obtenir certains bits d'informations en observant les séquences d'instructions CPU, comme les boucles et les branchements conditionnels. Donc cette méthode consiste à retrouver des informations sensibles, notamment les données liées aux clés cryptographiques, et ce, par l'analyse de la courbe représentative de la consommation de courant lors de l'exécution d'une commande. Cette courbe de consommation est différente suivant les instructions exécutées et les données manipulées.[38]

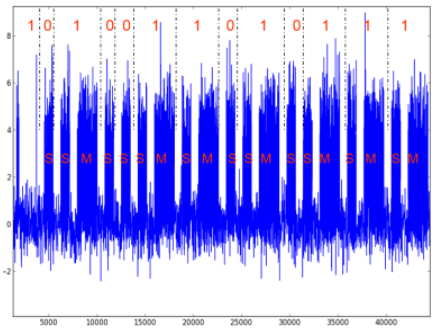


Figure 24- *Attaque SPA par simulation d'un circuit exécutant un algorithme [38]*

- **Attaques de type DPA pour Differential Power Attack** : Les attaques différentielles utilisent l'analyse statistiques et des techniques de correction d'erreur pour extraire les informations relatives aux clés secrètes. Elle nécessite moins d'informations sur l'implémentation de l'algorithme de la part de l'attaquant. Par ailleurs, cette technique gagne en performance par l'utilisation d'analyses statistiques pour aider à retrouver des informations des canaux auxiliaires. [39]

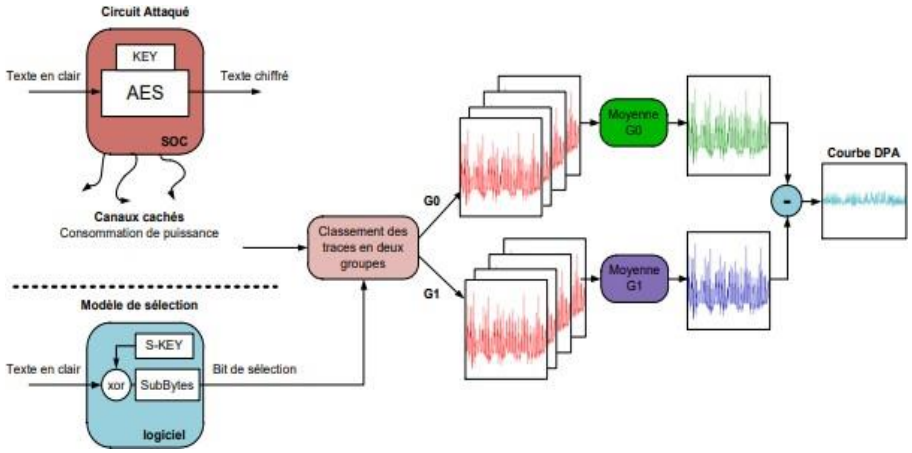


Figure 25- *Principe de l'attaque DPA [39]*

- Contrairement aux attaques passives, les attaques actives nécessitent de modifier certains signaux. Elles sont basées sur ***la perturbation*** des entrées/sorties et qui offre de nombreuses possibilités d'attaques et d'accès aux données sensibles. En appliquant des valeurs hors normes de façon continue ou transitoire, il est possible de perturber le microprocesseur et de modifier sélectivement une partie de la mémoire, des registres ou bien de provoquer un décodage erroné d'une instruction. (*Ex : Attaques par injection de fautes*).

Parmi les attaques basée sur la perturbation les plus répondus et les plus connus, on cite l'attaque par injection de fautes qui consiste à appliquer une force extérieure sur le matériel afin d'induire des erreurs ou fautes dans le microprocesseur. Ces fautes permettent dans une certaine mesure à l'attaquant de pouvoir obtenir et accédé à des informations et données sensibles stockées sur la puce, effectuer et achever des traitements normalement sécurisés par la réalisation d'un évitement du test d'un code PIN, ou encore d'essayer d'augmenter le nombre d'essai qui est généralement limité à trois (03) fois par le fabricant de la carte.

Pour réaliser une attaque par injection de faute, plusieurs méthodes et façon ont été exploitées depuis l'apparition des cartes à microprocesseur, et qui ont prouvés leurs succès, parmi lesquelles on cite ce qui suit :

- **Attaque électrique :** Cette attaque consiste à faire varier la tension d'entrée de l'alimentation de la puce d'une manière inattendue (*subitement et brusquement*), et ce, afin de perturber l'exécution de certaines opérations sur cette puce. Ce manouvre est suffisant pour obtenir une faute exploitable. En effet, la faille exploité est que la carte n'est pas auto alimentée l'énergie qu'elle utilise pour effectuer des opérations, et cette dernier lui vient du CAD (*lecteur*), le point qui rond possible l'exécution de cette attaque. *[34]*

- **Attaque par variation de fréquence de l'horloge :** En effet, la carte tire sa fréquence d'horloge du CAD (*lecteur*); donc la possibilité de faire varier celle-ci rond la carte vulnérable aux attaques de type injection de faute. En faisant varier la vitesse de l'horloge hors des limites autorisées par la carte, on peut induire des fautes au niveau du microprocesseur. Il est par exemple possible de faire doubler et changer la fréquence de l'horloge afin de faire exécuter au microprocesseur une opération (*OP2*), avec des paramètres qui devraient normalement être utilisés pour une opération (*OP1*) qui lui est antérieure. Donc l'attaque par variation de fréquence de l'horloge a exploité le même principe que l'attaque électrique. *[35]*

- **Attaque optique ou par laser :** Le principe de ce type d'attaque c'est d'utiliser une source lumineuse concentrée sur la puce et apporter suffisamment d'énergie à une cellule mémoire, pour changer son contenue. D'après les recherches qui ont été faites sur ce domaine, le matériel nécessaire à cette attaque n'est pas forcement coûteux c.-à-d. un flash d'appareil photo, ou un laser est suffi pour atteindre l'objectif. Généralement, après la réalisation d'attaque en utilisant cette technique, il est impossible de réutiliser la carte. Ainsi, on peut la classer parmi les attaques invasives. *[36]*
- **Attaque par perturbation électromagnétique** : Les états des cellules mémoires sont basés sur les principes des ions. Physiquement, la création d'un fort champ électromagnétique à proximité d'une cellule mémoire, va bouger les ions représentant l'état de cette cellule mémoire bougent, qui permettant ainsi de modifier le contenu de cette mémoire. Donc ce type d'attaque est basé sur la génération d'un champ électromagnétique à proximité de la carte à puce pour pouvoir perturber l'état de sa mémoire.
  Aujourd'hui, la majorité des cartes sont équipées de contremesures pour lutter contre l'injection de fautes par le courant électrique ou par la fréquence de l'horloge alors qu'il est encore difficile de lutter contre les attaques électromagnétiques ou optiques, car elles ajoutent de l'énergie directement à la surface de la puce. Ce sont donc les deux catégories d'attaques contre lesquelles il est le plus difficile de se protéger.*[37]*

L'objectif des attaques basées sur les techniques de perturbations des entrées/sorties c'est de causer une insertion ou modification ou exécution d'une faute instruction, entrainent des modifications et des perturbations dans le microprocesseur (exemple les registres contenus dans le microprocesseur), les différentes mémoires (Exemple code des applications, les clefs cryptographiques, le code PIN, objets temporaires manipulés, etc.), les bus (Modification des données qui transitent sur le bus).

2.1.2.2.Attaques logiques

Au début, les attaques matérielles qui ont ciblés les cartes à puce nécessitaient généralement des moyens physiques sophistiqués et des outils adaptés (*lasers, oscilloscopes, etc*.). Ces exigences réduisent par leur sophistication et leur cout le nombre d'attaquants basées sur la partie physique ou matérielle de la carte à puce. Par contre, de nouvelles attaques dite attaques logiques ont commencé depuis l'utilisation des applications et des systèmes d'exploitations embarqués, notamment ceux liées à la technologie Java Card. ***Ces types d'attaques reposent souvent sur la découverte d'une potentielle faille et généralement ne nécessitent aucun matériel sophistiqué mais elles deviennent plus difficiles à réaliser en vue les nouvelles mesures de sécurités implémentés sur ces systèmes.*** *[12]*

Les attaques logiques ciblant les cartes à puce consistent à attaquer la partie logicielle de la plateforme (*systèmes et applications*), et ce, par l'utilisation de plusieurs techniques et méthodes qui ciblent en particulier cette partie, à savoir : l'injection de données ou le chargement des applications malicieuses dans le but de contourner l'exécution des applications installées dans la carte à puce ou de dévoiler ses secrets (*clés cryptographies, code Pin, données sensibles, etc*.). Généralement les attaques logiques ciblant la partie logicielle englobent deux types d'attaques : les attaques qui exploitent les failles algorithmiques dues soit au non-respect de la spécification, et les attaques par chargement d'applications malicieuses dans la carte à puce.

Comme on a vue précédemment que parmi les mécanismes de sécurité offerts par la technologie Java Card on cite le vérifieur de bytecode et le pare-feu, ainsi le typage du langage Java. Ces mécanismes offrent, entre autre, des propriétés d'isolation forte des applications (*applets*) vis-à-vis de l'exécution de la machine virtuelle Java Card. En revanche, des limitations de cette isolation a été bien constaté, et ce, dû à l'évolution des attaques logicielles. Dans ce contexte, la Java Card peut héberger et loger des différents applets issues de divers fournisseurs et constructeurs. En outre, ces fournisseurs ne suivent pas, nécessairement, les mêmes standards de sécurité lors du développement de leurs applications. Par exemple, les conditions de sécurité imposées pour une application financière sont plus strictes que celles d'une application de divertissement.

Les premières attaques logiques sur la plate-forme Java Card ont été basées généralement sur deux approches intéressantes : une approche qui exploite un bug d'implémentation du «pare-feu» et une autre approche utilisant une confusion de type dans une applet mal formée. Plusieurs techniques ont été révélé pour faire une attaque de type logique sur la plate-forme Java Card, comme exemple l'obtention d'un accès non autorisé à la mémoire de la carte, qui est basée sur des techniques de confusion de type telles que la référence au tableau d'octets, la commutation de références d'objets différents et la création de tableaux en créant les objets avec la même représentation mémoire que le tableau désiré.

Ci-dessus, on cite quelques exemples d'attaques logiques les plus répondus et les plus connus et qui ont ciblé le coté logiciel des cartes à puce. La majorité de ces exemples ciblent les conteneurs de sécurité :

Dans la plateforme Java Card, les API Java Card fournissent et offrent plusieurs classes pour les données sensibles existes sur la puce, notamment les sous-classes Key pour le stockage des clés cryptographiques et la classe OwnerPIN pour les objets PIN, qui implante la fonctionnalité "Personal Identification Nimbera" et fournit la possibilité de mettre à jour ce code. La plate-forme doit garantir des mesures de sécurité pour ces objets contre les attaques logiques et/ou physiques, et ce, pour protéger l'intégrité et la confidentialité des codes PIN et des clés secrètes utilisées dans les processus de cryptographies. Des méthodes d'attaques ont été exploités par les attaquants pour accéder à ces informations (*Clés et code PIN*), enparticulier, l'utilisation de la fonctionnalité de décryptage que de la plate-forme Java Card elle-même, afin que les attaques logiques puissent toujours récupérer les clés et les codes PIN en clair.*[40]*

Autres méthodes ont été basées sur l'utilisation et l'installation d'une applet malveillante pour obtenir un accès illégal au contenu de conteneurs de sécurités d'une autre applet victime, et ce, pour obtenir les clés cryptographiques et le code PIN en clair. Cette pratiques est utilisée même pour réinitialiser le compteur d'essai de code PIN, qui permet par force brute de retrouver les codes secrets.

pin = new OwnerPIN(PIN_NB_LIMIT,MAX_PIN_SIZE); pin.update(bArray, bOffset, bLength);	Exemple de code lié à la création de l'objet PIN et l'intégration de la méthode update () qui affecte une nouvelle valeur au PIN et initialise le nombre de tentatives de PIN avec le nombre limite PIN_NB_LIMIT.

- **Attaque par modification du compteur d'essai de code PIN**

La classe API OwnerPIN offre aux développeurs d'applets une implémentation standard pour les objets PIN, qui stockent les codes PIN, puis fournissent toutes les fonctionnalités associées, y compris la gestion du nombre de tentatives autorisées pour prévenir les manœuvres frauduleuses. L'implémentation de l'API peut prendre en compte les faiblesses ou les contremesures existantes dans la plate-forme, y compris en ce qui concerne les attaques physiques. En effet, les spécifications de l'API indiquent que l'implémentation de la classe OwnerPIN doit être sécurisée contre certains types d'attaques, y compris les attaques qui tentent d'abuser du mécanisme de transaction. *[41]*

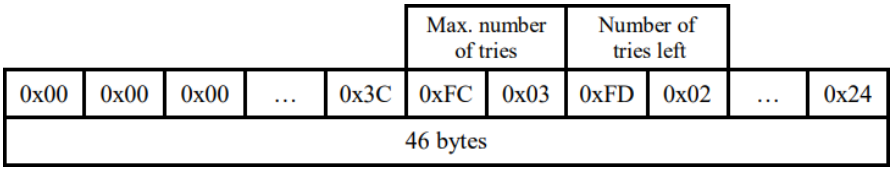


Figure 26*- Représentation de l'objet Owner PIN en mémoire. [40]*

L'attaque basée sur la modification du compteur d'essai consiste en l'utilisation d'une applet malveillante pour accéder à un emplacement de mémoire arbitraire en manipulant des métadonnées d'objets de tableau. Il est possible donc d'observer la représentation d'un Objet OwnerPIN en mémoire brute (*Figure 26*). Il est clair que sue la plate-forme Java Card, il est difficile d'accéder au code PIN, mais il est possible de voir les octets dans la structure de données ***OwnerPIN,*** et qui enregistre le nombre maximum d'essais autorisé et le nombre d'essais restants. Donc par la réinitialisation du compteur d'essai aux valeurs maximales, il est possible de deviner le code PIN par plusieurs répétitions (force brute).

- **Attaque par récupération des clés DES**

Pareille que la gestion du code PIN, l'API Java Card fournit également des classes pour stocker, manipuler et utiliser des clés cryptographiques, par exemple : la classe *DESKey* pour les clés DES. Les méthodes de cette classe incluent *setKey (byte[] keyData,short kOff)* et *getKey (byte[] keyData,short kOff)* pour lire et écrire la valeur de la clé. Pour effectuer ce typed'attaque, l'applet malveillante lit la représentation en octets bruts d'une clé DES à partir de la mémoire d'une autre applet et la copie dans un objet clé DES qui lui est propre. Une fois la clécrypté est copiée, alors, l'utilisation de la méthode *DESKey.getKey()* par l'aplet malveillante permet la récupération du texte clair de la clé.

Le contenu de l'objet clé de l'applet victime sera copié dans un objet appartenant à l'applet malveillante, car le pare-feu de la plateforme Java Card empêche l'applet malveillant d'invoquer la méthode *getKey()* sur un objet Key appartenant à une autre applet.*[41]*

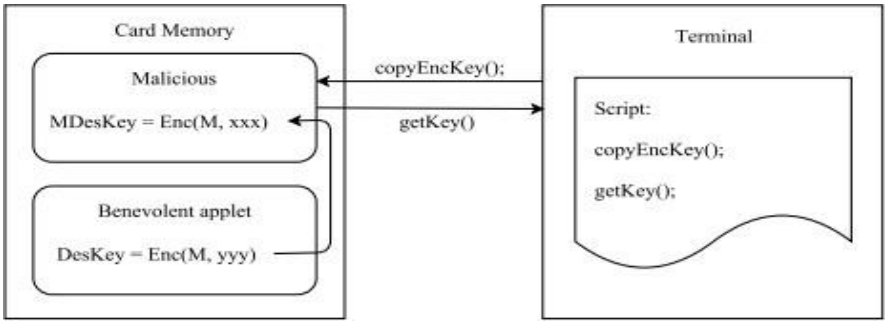


Figure 27*- Récupération des clés DES en clair [42]*

- **Attaque par récupération du code PIN en clair**

Il est clair que La classe *OwnerPIN* ne fournit pas de méthode *getPIN(),* similaire à *getKey()* de la classe *DESKey*, pour récupérer les codes PIN en clair. Doc Si une telle méthode *getPIN()* était disponible, nous pourrions annuler le cryptage des codes PIN de la même manière que nous avons défait le cryptage des clés DES comme on a vue précédemment, en copiant la représentation cryptée des octets du code PIN vers un autre objet PIN et en invoquant *getPIN().*

Sachant que le contenu du code PIN et les objets des clés sont stockés cryptés sur la carte, les attaquants ont pensé de la possibilité que le même algorithme est utilisé dans les deux cas. Cela conduit à une attaque où l'opération *getKey()* de la classe *DESKey* soit utilisée pour récupérer les codes PIN en clair. Dans ce cas l'applet malveillante copie la représentation chiffrée d'un PIN appartenant à une autre applet à l'un de ses propres objets clés, puis appelle *getKey()* pour récupérer le texte en clair. Comme contremesure qui a été déployé pour contrecarrer cette attaque c'est l'utilisation d'une clé différente pour chiffrer les codes PINs.*[41]*

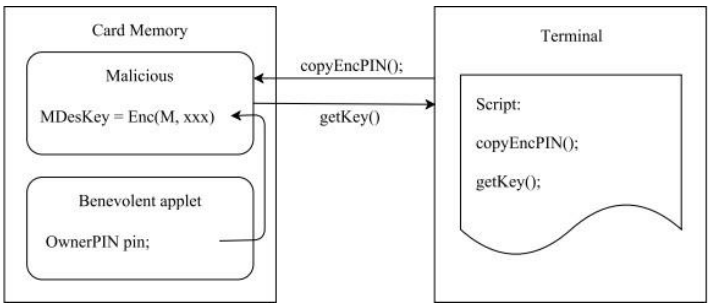


Figure 28- *Récupération du code PIN en clair (Owner PIN attaque)*

- **Attaque par accès illégal à la référence du tampon APDU**

Comme on a vue précédemment, parmi les composantes de la sécurité déployée dans la technologie Java Card, la fonctionnalité de pare-feu. Dans ce cadre, l'environnement d'exécution Java Card restreint le stockage de références à des tableaux globaux, car ils sont potentiellement partagés entre les applets. La spécification de l'environnement d'exécution indique que tous les tableaux globaux sont des objets de tableau global temporaires. Ces objets sont la propriété du contexte JCRE, mais peuvent être accédés à partir de n'importe quel contexte. Cependant, les références à ces objets ne peuvent pas être stockées dans des variables de classe, des variables d'instance ou des composants de tableau. Le stockage de références à ces objets, donc, sont restreint au pare-feu, et ce, pour empêcher toute réutilisation non autorisée.

L'idée de l'attaque c'est d'essayer de contourner la restriction sur le stockage des références aux objets globaux par une applet malformée installée sur une carte qui ne dispose pas d'un vérifieur de bytecode sur carte.*[41]*

- **L'attaque par confusion de type**

La confusion de type consiste à utiliser deux références de types différents (et incompatibles), exemple type 'short' et type 'byte, et ce, pour accéder à la même zone physique de la mémoire. L'accès à un tableau de byte comme étant un tableau de short permet de lire les données au-delà des limites du tableau de byte. Plus exactement, ça va engendrer la lecture de deux fois la taille du tableau de byte de données et qui sortent potentiellement du domaine de l'applet. Donc, possibilité d'accéder à des zones de la mémoire non autorisé.

La technique de confusion de dans la machine virtuelle Java qui s'exécute sur la carte, signifie tromper la JVM en créant deux références de deux différents types pointant vers le même emplacement mémoire.

Par exemple (figure 29) si le bloc de mémoire a été à l'origine alloué en tant que tableau byte. L'accès à ce bloc comme un tableau short donne la possibilité de lire les données au-delà des limites légales du tableau byte, donc avoir un accès à une zone mémoire normalement inaccessible au-delà du tableau byte. *[42]*

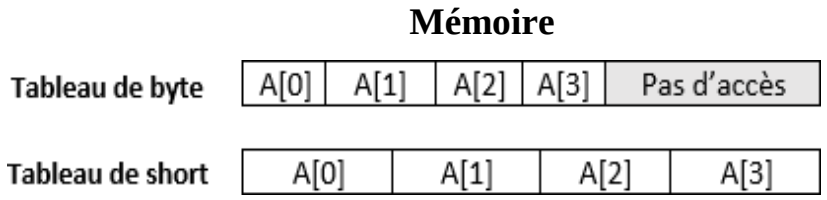


Figure 29- *Confusion de type entre tableau [08]*

2.2.Attaques externes

Les attaques citées précédemment s'appuient sur un accès sur la carte à puce elle-même, que ce soit sa partie physique ou sa partie logique. En revanche, les attaques que nous allons aborder dans cette partie ciblent les transactions électroniques effectuée dans le contexte d'une carte non présente (*les transactions en ligne*), en s'appuyant sur un accès logique soit au niveau d'équipements utilisés par du le porteur de la carte (*Ordinateur, smartphone, Tablette, etc.*), soit au niveau les serveurs distants (*De la banque émettrice de la carte ou du commerçant*).

Ces attaques ont pour objectif de collecter les informations sensibles de la carte pour les utiliser dans des transactions frauduleuses en contournant les mécanismes d'authentification et d'identifications mises en place, à savoir système d'authentification basé sur le protocole 3DSecure ou encore le plus récent le protocole 3Dsecure2 ou EMV-3Dsecure.

2.2.1.Les attaques ciblant le porteur de la carte

En effet, l'appareil utilisé par le porteur de la carte pour effectuer une transaction de type carte non présente est considéré comme le maillon le plus faible dans la chaine qui constitue ce type de transaction électroniques, parce que généralement l'appareil du porteur de la carte est exclu du périmètre de système de sécurité appliquée.

Dans cette partie nous mettons en lumière les différentes attaques visant l'appareil du porteur de la carte, qu'ont été classées comme efficace contre les mesures de sécurité adoptées actuellement dans notre pays. Nous avons classé ces attaques en basant sur l'impact de ces attaques sur l'appareil de la victime, et aussi sur les tendances actuelles et les méthodes utilisées par les cybercriminels.(figure 30)

Les cybercriminels exploitent les vulnérabilités inhérentes aux personnes (*ingénierie sociale et hameçonnage*) afin de prendre le contrôle de leurs appareils (*Généralement par le déploiement des malwares et logiciels malveillants*), et vol d'informations d'identifications et d'authentification utilisées par les porteurs légitimes de la carte. Ces manouvres sont généralement terminées dans des faux pages web redirigés par les malwares déjà déployés.

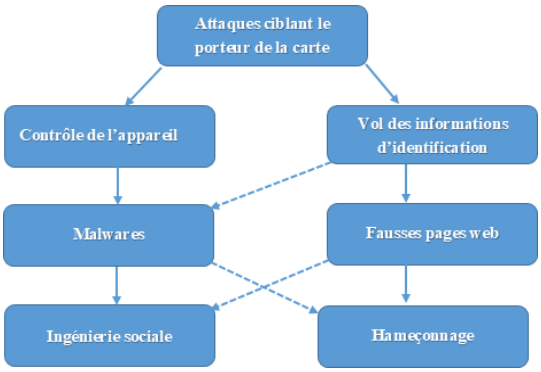


***Figure 30**- les attaques ciblant le porteur de la carte. [43]*

2.2.1.1.Vol des informations d'identifications :

Le vol d'information d'identification et d'authentification est l'attaque la plus répandue et la plus connu dans le domaine de la fraude liée à la transaction sans carte. Elle consiste principalement à obtenir les informations confidentielles de la carte et aussi du porteur de cette carte, à savoir : le numéro de carte, la date expiration, le cryptogramme visuel, le nom du porteur, numéro de téléphone, etc. Ces information sont nécessaires pour pouvoir effectuer des transactions et sert, entre autre, d'authentifier l'utilisateur légitime, en fournissant des informations qui devront normalement être connues seulement par le porteur de la carte.

Actuellement, cette technique est utilisée contre les mécanismes de sécurité telle que l'authentification a doublé facteur, où l'attaquant doit capturer toutes les informations nécessaires pour reproduire un accès légitime. Généralement, l'attaque se résume aux étapes suivantes :

- L'attaquant construit un logiciel malveillant ou un site web similaire à ce de la banque émettrice de la carte de la victime.
- En appliquant les méthodes de l'ingénierie sociale l'attaquant parvient à installer le logiciel malveillant sur l'appareil de la victime, comme il peut parvenir a invité la victime a visité le site web contrefait afin d'indiquer les éléments d'authentifications ou de paiement.
- Lorsque la victime accède au service de la banque, les informations d'authentifications seront capturées par le logiciel malveillant ou par la page web contrefaite.
- Une fois les informations nécessaires à l'authentification sont obtenues, la victime sera rediriger d'une manière transparente vers le système légitime.
- Apres avoir obtenu les données nécessaires, l'attaquant peut accéder au système en tant que utilisateur légitime.

Les authentifications fortes basées sur la réception du SMS comme la 3DS sont ainsi tout particulièrement fragilisées, parce que l'attaquant peut procéder au changement du numéro de téléphone portable de la victime (*sur le site internet de la banque ou en contactant l'organisme financière*). Plusieurs mécanismes ont été, aussi, exploitées pour contourner les systèmes d'authentification moderne.

2.2.1.2.Contrôle de l'appareil du client :

L'objectif de cette attaque est d'avoir le contrôle total sur l'appareil de la victime au lieu de voler seulement les données utilisées dans le processus d'identification et d'authentification. Ce manouvre se réalise par l'installation des logiciels malveillants sur les appareils cibles, ensuite l'appareil compromise est utilisé pour accéder aux services souhaités à savoirs les

services de la banque en ligne (*Paiements électroniques, virements, transferts, etc*.). Cette attaque est plus complexe à réaliser par rapport un vol d'informations par des simples techniques d'hameçonnage. Ce type d'attaque se résume dans les étapes suivantes : *[44]*

- L'attaquant construit et publie des logiciels malveillants.
- En exploitant des vulnérabilités dans des applications, ou par l'utilisation de l'ingénierie sociale, l'attaquant parvient à installer ledit logiciel malveillant sur l'appareil de la victime.
- Apres avoir un contrôle sur l'appareil, l'attaquant utilise le malware pour accéder au système bancaire en tant qu'utilisateur légitime, puis il effectue des transactions frauduleuses.
- Tant que le malware est actif sur l'appareil de la victime, l'attaquant peut supprimer les transactions frauduleuses effectuées (historique d'applications bancaire ou de navigateur, SMS etc.), afin d'empêcher la détection de la fraude ou l'attaque par la victime.

2.2.2.Les attaques ciblant les serveurs distants (Banques ou E-Marchants)

2.2.2.1. Les attaques e-skimming

Les banques ou les commerçants peuvent également faire l'objet d'attaques sur leurs systèmes d'information. Malgré que sont particulièrement protégés, ces systèmes sont une cible de choix pour les fraudeurs qui peuvent espérer compromettre un nombre important de cartes de paiement. Actuellement, les attaques les plus connu qui visent de plus ces systèmes sont connu sous «les attaques Magecart» ou «Web skimming attacks » en anglais, qui visent principalement les sites d'E-Marchants, pour détourner les données bancaires.

L'objectif principal de cette attaque est toujours le même : installer sur le site de la victime un script malveillant, souvent désigné sous le terme de «skimmer», en anglais qui se chargera sur la page de paiement. Celui-ci vise le plus souvent à imiter le formulaire de recueil des informations d'authentifications afin de substituer une version malveillante qui transmettra aux attaquants les codes volés, tout en poursuivant la transaction. En Algérie les informations Bancaires liées aux clients ne sont pas stockées au niveau des sites E-Marchants, en revanche des manouvres pareilles peuvent cibler les organismes financiers tels que les banques ou intermédiaires tel que la SATIM.

2.2.2.2. Les attaques à force brutes (les attaques BIN)

L'abréviation BIN signifie le numéro d'identification bancaire, qui est constitué des quatre ou six chiffres premiers d'un numéro d'une carte de crédit, Ces numéros sert a identifié la banque émettrice de la carte, Les autres numéros sont uniques à chaque titulaire de carte.

Le principe de fonctionnement de cette technique d'attaque est de garder les six premiers numéros de la carte et de générer le reste des numéros par <moulinage >, puis l'attaquant essaye les différentes combinaisons possible de la date de validation ainsi que le cryptogramme visuel, sur des sites de e-commerce tel que le site du géant américain de commerce électronique 'Amazon'. Une fois qu'une combinaison valide est obtenue, l'attaquant peut effectuer des achats importants rapidement. *[46]*

2.3.Tableau récapitulatif des attaques

Dans le tableau suivant, nous résumons les différents types d'attaques ciblant les cartes à puces (Cas de la carte bancaire) par classe d'attaque en indiquant aussi le cout de faisabilité de chaque type d'attaque ainsi que certaines contremesures existantes pour chaque classe d'attaque.

Classe d'attaque			Attaque	Cout de faisabilité	Description	Contremesures
Interne (carte présente)	Physique	Invasive	• L'attaque de l'Etching ; • Scanning Electron Microscopes (SEM) ; • Probing ; • FIB (Focused Ion Beam)	Très élevé	Ce sont des attaques extrêmement performantes nécessitant un accès physique à la carte ainsi que du matériel important (Station d'analyse, SEM, FIB, Microscope, Laboratoire chimique, etc.) ce que les rendent couteuse sur les deux plans matériels et temps.	• Conception de blocs aléatoires ou brouillés ; • Bouclier métallique ; • Détecteurs et indicateurs d'anomalies ; • Structure multicouches et miniaturisation.
		Non-invasive	• Attaque SPA ; • Attaques DPA ; • Attaque électrique ; • Attaque par variation de fréquence de l'horloge ; • Attaque optique ou par laser ; • Attaque par perturbation électromagnétique.	Très élevé	Ces attaques consistent à l'analyse d'informations provenant de la carte sans affecter son intégrité, toutefois ces attaques nécessitent des connaissances techniques importantes et la possession des outils d'analyse les plus récents.	• Exécution constante ; • Retards aléatoires ; • Dispersion des informations ; • Exécutions aléatoires ; • La suppression de l'interfacede test ; • La mise en place des Solutions contre l'injection de fautes (compteur de faute, redondance de calcul).
	Logique		• Attaque par modification du compteur d'essai de code PIN ; • Attaque par récupération des clés DES ; • Attaque par récupération du code PIN en clair ; • Attaque par accès illégal à la référence du tampon APDU ; • L'attaque par confusion de type.	Elevé	Ces attaques exploitent des failles pour contourner les protections mises en place, généralement, il s'agit d'une mise à défaut des mécanismes d'isolation et d'intégrité du code et des données des applets, ce type des attaques nécessitent des connaissances techniques importantes dans en cryptographie et en développement des applets.	• Conception structurée du logiciel ; • Utilisation des modèles mathématiques (RSA, le modèle aléatoire d'oracle) ; • L'application des mécanismes de contrôle d'accès à la mémoire ; • L'amélioration des schémas cryptographique ; *[32]*
Externe (carte non présente)	Coté Client		• Vol des informations d'identifications ; • Contrôle de l'appareil du client.	bas	Ces attaques exploitent les vulnérabilités inhérentes aux personnes (ingénierie sociale et hameçonnage) afin de prendre le contrôle de leurs appareils (malwares), et vol d'informations d'identifications et d'authentification.	• Certificats numérique ; • Protection du navigateur ; • Utilisation des jetons OTP ; • Captcha ; • Services des messages courts ; • Identification de l'appareil ; • Clavier virtuel.
	Coté serveur distant		• Les attaques e-skimming ; • Les attaques à force brutes (les attaques BIN).	élevé	Ces attaques ciblent principalement Les sites des banques et les commerçants afin d'obtenir un volume important des informations des cartes de paiement.	• Audit des ressources Web ; • Utilisation de la surveillance et l'inspection automatisées ; • Utilisation des bibliothèques JavaScript et des scripts tiers sûres ; • utilisation des pratiques de développement logiciel sécurisées ; • utilisations des solutions IDS.

Tableau récapitulatif des attaques ciblant les cartes et les données bancaires

Conclusion

Plusieurs sources internationales ont confirmé que la majorité de la fraude et d'attaques qui ciblent actuellement les moyens de paiement (carte à puce) dans le monde se font en ligne, sans la présence de la carte elle-même (Attaques qui ciblent le porteur de la carte ou ceux qui ciblent les serveurs distants des Banques). Dans la partie pratique de ce travail (Chapitre 2), on s'intéresse à une simulation d'attaque qui cible les données d'identification et d'authentification pendant une transaction en ligne (Carte non présente) dans le contexte algérien.

Les attaques contre les systèmes d'authentification et d'identification consistent à utiliser l'ensemble d'outils et de techniques par les attaquants pour contourner les mécanismes d'authentification utilisées dans les systèmes de paiement actuels, notamment le système d'authentification basé sur le protocole 3DSecure ou encore le plus récent le protocole EMV-3Dsecure. Ces techniques sont utilisées quel que soit le facteur d'authentification utilisé, qu'il s'agisse d'un mot de passe unique OTP envoyé par SMS ou de notifications poussées envoyées vers une application mobile. D'après des organismes spécialisés dans l'analyse des phénomènes liés à la fraude monétiques, les méthodes qui permettent de contourner l'authentification sont actuellement les plus exploitées et sont très efficaces dans le processus de contournement des mécanismes de sécurité mises en place.

CHAPITRE II
Conception d'une attaque sur les cartes à puces (Cas de carte Non-Présente)

Chapitre II : Conception d'une attaque sur les cartes à puces

Introduction :

Plusieurs améliorations en termes de sécurité ont été apportées aux systèmes basées sur les cartes à puces ces dernières années. Ces améliorations ont permis de diminuer le taux de fraude et d'attaques qui ciblent l'intégrité de ces cartes, et ce, par le développement des contremesures et des mécanismes de défenses contre ces types d'attaques que ce soit au niveau logique ou au niveau physique *(Attaques matérielles et attaques logicielles*). D'après la Banque Centrale de l'Europe, la fraude monétique liées aux moyens de paiement électroniques à savoir les cartes bancaires ainsi que les terminaux de paiement, a diminué à 56.8% au cours des cinq années de la période (2015-2019). *[47]*

En 2019, le taux de la fraude ciblant les données des cartes bancaires durant une transaction en ligne (*transaction à base de carte non présente*) représentait 80% de la valeur totale de la fraude à la carte *[48]*. Ce taux de fraude lié aux transactions en ligne ne cesse pas à se croitre régulièrement, malgré les mécanismes de sécurité misent en place afin de sécuriser ce type des transactions. Pour cela nous s'intéressons aux attaques ciblant les transactions effectuées dans le contexte de carte non présente.

Comme nous avons indiqué dans le chapitre précédent, la sécurité des transactions en ligne par carte bancaire (carte à puce) se repose sur le protocole 3D-Secure. Ce protocole a été introduit par l'organisme EMVCO et adopté par plusieurs réseaux de carte de bancaire (Cas de la SATIM en Algérie). Le fonctionnement de ce protocole consiste à rediriger le navigateur de titulaire de la carte bancaire vers le site web de la banque émettrice de la carte, ou le titulaire de la carte s'authentifié en introduisant les informations de sa carte puis un code à usage unique OTP (*One Time Password*), reçu généralement par un SMS, afin de valider la transaction et authentifier le porteur.

L'objectif des simulations réalisées de notre part est de montrer la faiblesse du protocole 3DS et comment ce dernier peut être contourné en utilisant une attaque basé sur l'hameçonnage traditionnel, l& technique de main in the middle (MITM), et un logiciel espion.

1 Description générale

Dans la simulation des attaques nous allons passer par trois (03) scénarii indépendants. (Tableau 3)

Le premier scénario consiste à effectuer une transaction frauduleuse en usurpant les informations de la carte bancaire ainsi que le code de vérification (*OTP*) de la transaction reçue par SMS par la victime. Pour cela, nous allons utiliser un hameçonnage traditionnel. Dans le deuxième scénario nous allons utiliser un hameçonnage Man-in-the-Middle (MitM). Le but de cette étape est pour procéder au vol des informations de connexion au compte en ligne (*ex : compteclient ou compte bancaire de la victime*), dont l'objectif est de changer des informations pertinentes comme le numéro de téléphone de la victime. Ce manouvre nous permet soit d'effectuer d'autres transactions frauduleuses soit de mieux exploitées les informations obtenues dans le premier scénario. Dans ce tableau qui suit, nous détaillons la différence entre ces deux types d'hameçonnage utilisée.

Dans le troisième scénario, nous allons transformer une application légitime d'un organisme bancaire ou institution financière en une application malveillante. Cette application malveillante sera transmise à la victime par l'exploitation de plusieurs techniques et méthodes basées sur l'ingénierie sociale, le phishing (Mail, SMS, déployer sur google playStore, etc.). Cette dernière permet de dérober des données sensibles.

	Hameçonnage traditionnel	Hameçonnage Man-in-the-Middle
Méthode	L'attaquant crée une image du site Web cible à l'aide d'un outil de clonage ou manuellement.	Un serveur proxy inverse agit comme MITM en interceptant dynamiquement les requêtes du client et initier une nouvelle connexion de l'attaquant vers le site Web cible.
Information recueille	Username, passoword, réponse à une question secrète.	username, passoword, réponse à une question secrète, code reçu par MFA session cookies.
Avantage pour les attaquants	Facile à mettre en action.	Difficile à détecter et capable de surmonter l'authentification a deux facteurs.
Inconvénient pour les attaquants	L'existence des solutions pour détecter et arrêter les sites de phishing.	Nécessite des connaissances avancées pour mettre en place.

Tableau 3 : comparaison entre l'hameçonnage traditionnel et l'hameçonnage MITM

2 Scénario I : Hameçonnage Traditionnel

2.1 Description de l'attaque :

Dans cette attaque nous allons commencer typiquement par l'envoi d'un e-mail enusurpant l'identité d'un organisme bancaire. Le message avertit la victime qu'un problème est survenu avec sa carte bancaire, ce que nécessite une action de vérification immédiate de ces informations ainsi que du numéro de téléphone liée à la carte bancaire. La victime est ensuite dirigée vers une fausse page web similaire au site officiel de l'entité usurpée (*la banque à titred'exemple*), sur laquelle la victime est invitée à entrer les informations de la carte et données personnelles (*le numéro de la carte bancaire, la date de validité, le cryptogramme visuel, des informations personnelles, numéro de téléphone*, etc.).

Après, nous allons demander à la victime de confirmer son numéro de téléphone en renseignant le code à usage unique reçu par SMS, suite à une transaction frauduleuse que nous avons initiée par les informations usurpées auparavant. (Figure 31)

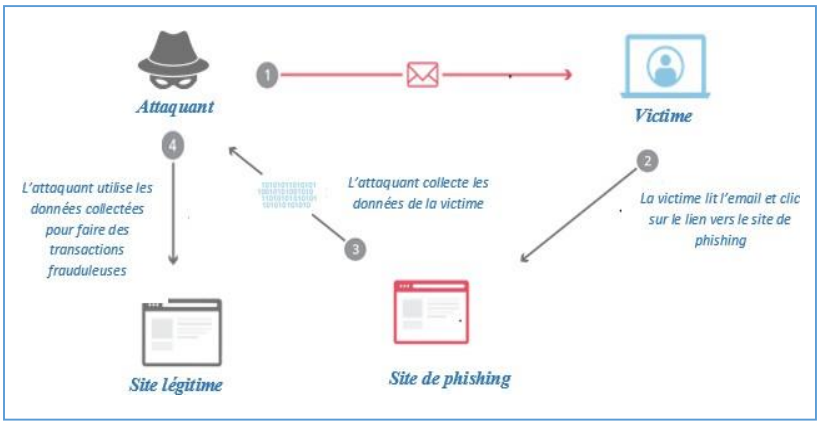


***Figure 31**- Schéma de l'hameçonnage traditionnel par mail.*

2.2 Préparation de l'attaque

Cette attaque basée sur un hameçonnage traditionnel se déroule en étapes suivantes :

2.2.1 Définition de la stratégie d'attaque

La stratégie d'attaque se résume en :

- Choix de l'entité bancaire à usurper (L'organisme financière).
- Choix du vecteur d'attaque (Email ou SMS à envoyer dans notre cas) et l'action attendue par lavictime (Suivre les instructions derrière un lien qui va rediriger la victime vers notre siteweb de phishing) ;
- Le mode d'exploitation des données collectées (dans notre cas les données usurpées vont être utilisées pour initier une transaction frauduleuse) ;
- Préparer les moyens d'atteindre les victimes : les listes d'emails et/ou de numéros de téléphones.

2.2.2 Création du support d'attaque

- Rédaction du contenu textuel conformément au scénario et construction du message à envoyer à la victime ;
- Création du site d'hameçonnage et la base de données pour le stockage desinformations usurpées ;
- Préparation et configuration de l'infrastructure (zones DNS, IP…) pour la mise enœuvre et le déploiement de l'attaque.

2.3 Diagramme de séquence de déroulement de l'attaque

Ce scénario se base principalement sur l'envoi d'un message (figure 32) frauduleux en sepassant par l'identité ou l'organisme bancaire de la victime (Usurpation d'identité). En premierlieu le choix du temps opportun pour l'envoi de l'e-mail est indispensable pour empêcher toutescontacte réel du client avec sa banque (Ex : les Week-End, les Jours des Fêtes, etc.).

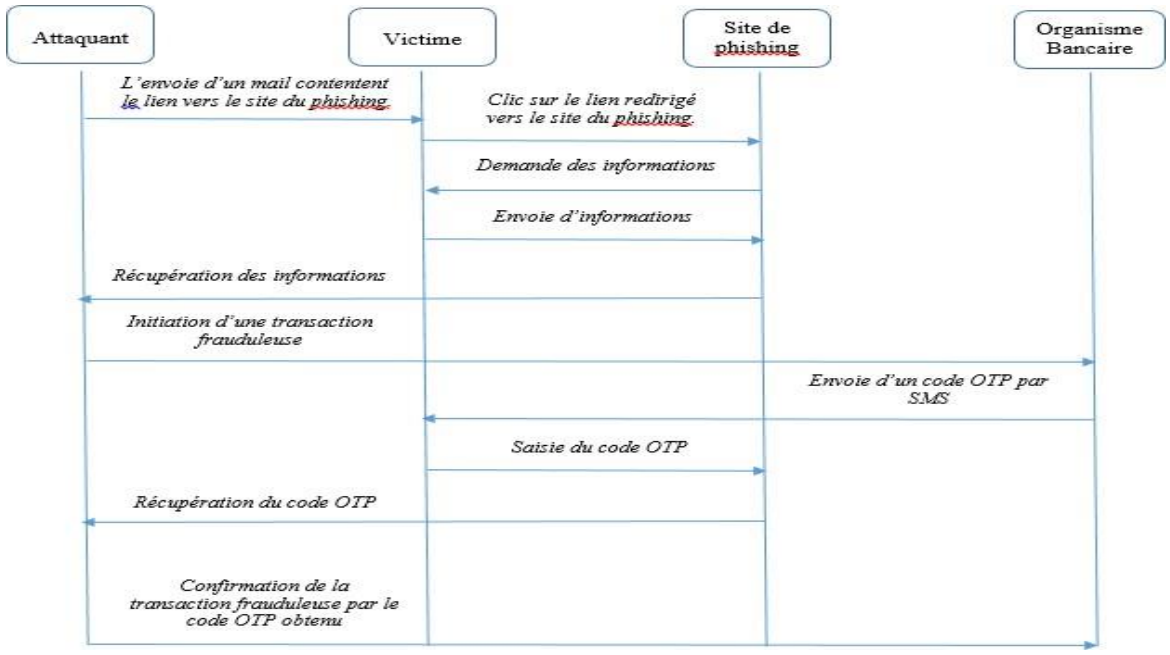


***Figure 32**- Diagramme de séquence de l'attaque par hameçonnage traditionnel*

Dans un premier temps, l'attaquant envoie à la victime l'e-mail frauduleux contentent le lien vers le site du phishing. La victime va ensuite exploiter le contenu envoyé, et procédera à la vérification immédiate de ces informations, en vue le caractère urgent de l'e-mail.

En réalité ce dernier est dirigée vers une fausse page web similaire au site officiel de sa banque, sur laquelle ce victime saisi les informations de la carte et données personnelles (le numéro de la carte bancaire, la date de validité, le cryptogramme visuel, des informations personnelles, numéro de téléphone, etc.).

L'attaquant va ensuit procédera à l'initiation d'une transaction frauduleuse et obtiendra le code à usage unique reçus par la victime de la même manière précédente (page de phishing inclue dans le même scénario). Cette transaction sera confirmée par l'attaquant en utilisant ce code OTP obtenu.

3 Scénario II: Hameçonnage Man-in-the-Middle

3.1 Description de l'attaque :

Ce scénario d'attaque est basé sur deux principes : l'attaque de l'homme du milieu (HDM) ou man-in-the-middle (MitM) et le phishing. L'idée générale est de rediriger la victime vers un proxy inverse qui va jouer un rôle intermédiaire entre cette victime et le site authentique d'où l'aspect HDM. Le proxy inverse va usurper le site authentique sollicité par la victime et collecte le flux de données ciblées, d'où l'aspect phishing.

Dans ce scénario, au lieu de cloner le site authentique (Dans notre cas le site de l'organisme bancaire), nous utilisons un proxy inverse, alimentant le contenu de l'utilisateur depuis le site réel pour rendre l'attaque plus convaincante et disperser les soupçons au niveau de la victime.

La victime reçoit un contenu authentique du site légitime, de telles sortes que tout le trafic et toutes les interactions de la victime avec le site légitime sont contrôlées par le serveur du reverse proxy, dès que la victime effectue une authentification a deux facteurs avec le site légitime. Le serveur recueille les informations d'identification en temps réel et aussi l'identificateur de session. Dès qu'elles sont récoltées, elles seront placées dans les fichiers journaux. Ces informations peuvent être utilisé en temps réels pour se connecter au compte de la victime afin d'effectuer des transactions frauduleuses ou aussi changer des informations sensibles comme le N° de téléphone lié au compte client. (figure 33)

Figure 33- Schéma du reverse proxy

3.2 Préparation de l'attaque

Les étapes de cette attaque déroulent comme suit :

3.2.1 Définition de la stratégie

La stratégie d'attaque se résume en :

- Choix de l'entité bancaire à usurper
- Choix du vecteur d'attaque (SMS dans notre cas) et l'action attendue (clic sur un lien redirigeant la victime vers le site web miroir) ;
- Le mode d'exploitation des données collectées (dans notre cas la session usurpées va être utilisée pour effectuer une transaction frauduleuse et modifier le numéro du téléphone de la victime) ;

3.2.2 Configuration du support d'attaque

- Choix d'un service d'envoie des SMS (pour envoyer le lien de site web miroir) ;
- Préparation et configuration de l'infrastructure (zones DNS, IP, machine virtuel Linux) pour la mise en œuvre de l'attaque ;
- Configuration du serveur reverse proxy.

3.3 Diagramme de séquence de déroulement de l'attaque

Le déclenchement de l'attaque commence par l'envoi d'un SMS contenant un lien du site malveillant, la victime hameçonnée accède à ce site en pensant qu'il s'agit du site légitime de l'organisme bancaire, et sera invité à saisir ses informations de connexion.

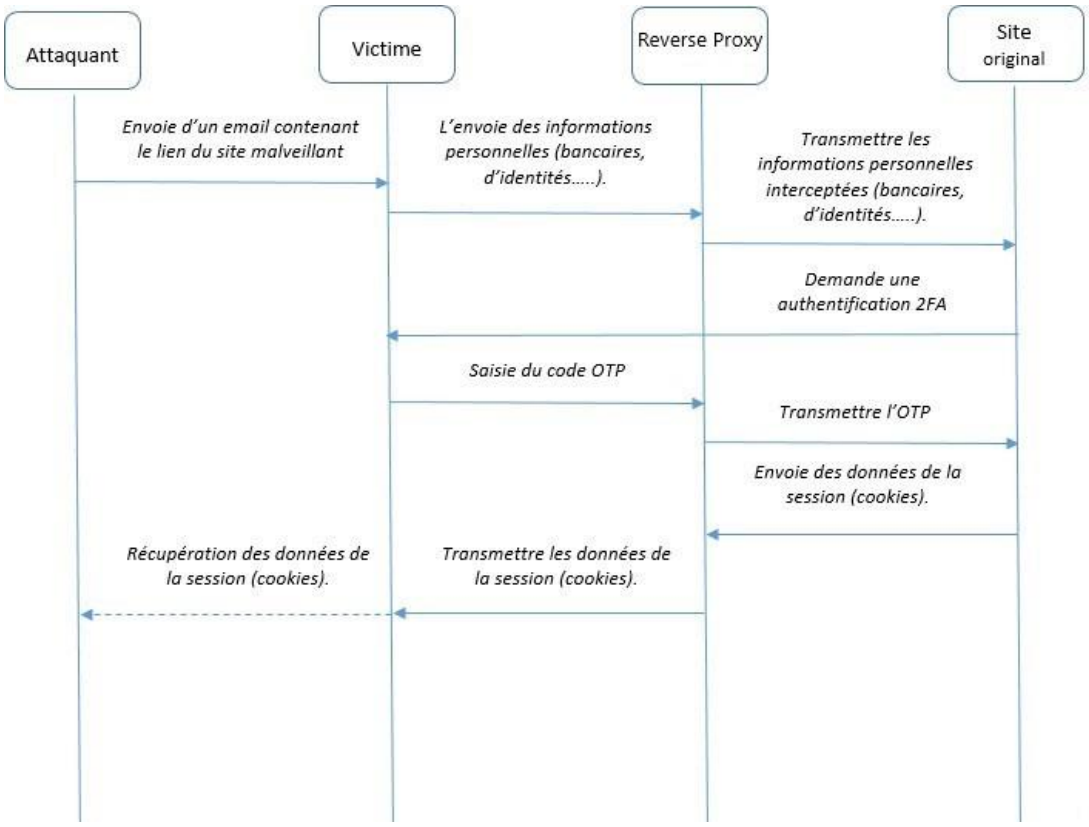


Figure 34- *Diagramme de séquence de l'attaque basée sur le reverse proxy*

Après avoir saisi leurs informations d'identification, la victime recevra le code de vérification de la part du site légitime. Dès que la victime valide l'authentification avec le code reçu, le reverse proxy extrait les données d'authentification de l'utilisateur, tout en conservant une copie de la session authentifiée. Dans cette situation, l'attaquant peut utiliser cette copie de session pour se connecter au compte du client et effectuer des transactions frauduleuses ou modifier le numéro de téléphone enregistré.

4 Scénario III : Vol des informations d'identification par logiciel espion

4.1 Description de l'attaque

Avec la popularité croissance d'Android au cours de dernières années, Android est devenu le plus populaire parmi les utilisateurs ainsi que les pirates à nos jours, Google Android détient 68.79% du marché mondial des systèmes d'exploitation mobiles *[50]*. Android est également devenu populaire entre les développeurs grâce à son SDK open source qui facilite le développement des applications mobile avec moins d'effort, ce qu'ont augmenté les activités des attaquants ciblant Android via des applications malveillantes. Le nombre des malwares ciblant Android dépassent les deux-cent (200) milles au cours de cette année 2023. *[51]*

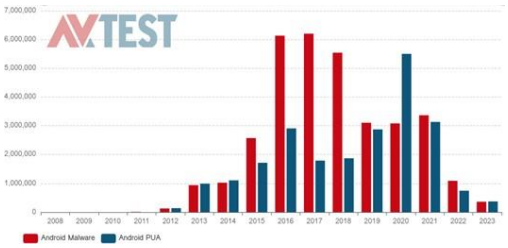


Figure 35- *Statistique sur les malwares et les payloads ciblant Android [56]*

De nos jours, la plupart des gens se servent de leur téléphone pour gérer leurs transactions financières. A cet effet, les smartphones sont devenus l'une des proies préférées par les pirates qui s'intéressent à commettre de la fraude monétique, en ceci en élaborant des malwares bancaires qui visent les smartphones des victimes afin d'usurper les informations de sa carte bancaire et ses données personnelles (le numéro de la carte bancaire, la date de validité, le cryptogramme visuel, nom et prénom du titulaire de la carte, numéro de téléphone, etc.).

Dans ce scénario d'attaque, nous allons transformer une application légitime d'un organisme bancaire en une application malveillante. L'application malveillante sera transmise à la victime via l'une des méthodes de phishing (Mail, SMS, déployer sur Google Play Store etc..). L'application installée sur le smartphone de la victime permet de dérober les informations de la carte bancaire ainsi les informations de connexion utilisées par l'application bancaires (Username, Password). Cette application sera aussi capable de contourner l'authentification à deux facteurs (2FA), car elle est en mesure d'accéder aux codes SMS et de les envoyer vers le pirate via le service Gmail.

4.2 Préparation de l'attaque

4.2.1 Désassemblage de l'application légitime

Dans cette étape nous allons utiliser une technique d'ingénierie inverse (Reverse engineering) qui consiste à désassembler l'application bancaire légitime (figure 36). Parmi les fichiers engendrés par l'opération de désassemblage, nous intéressons aux contenus dans le répertoire Smali, qui comprend tous les fichiers d'application écrits en Smali.Les fichiers .smali reflètent le comportement de l'application d'origine mais ont un défaut, à savoir que Smali n'étant pas un langage de haut niveau, il n'est pas immédiatement lisible.

IL est possible de changer et déboguer le code pas à pas en rajoutant des fonctionnalités cachées, et sur ce principe qu'on va entamer notre processus de développement. Le principe c'est de développer une application en se basant sur une autre application largement utilisées et exploitées par le public (Fonctionnalités malveillantes cachés dans une application légitime).

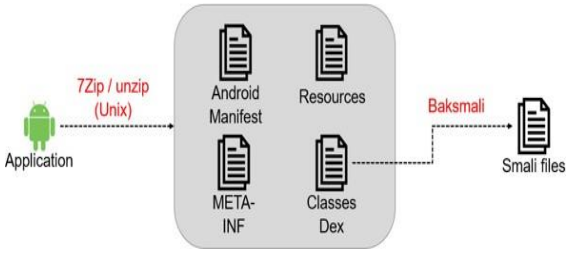


Figure 36- *Désassemblage d'une application Android.*

4.2.2 Préparation du logiciel malveillant :

Dans cette étape, on va développer une application Android basique contenant plusieurs codes malveillants. Cette application est chargée de faire plusieurs actions malveillantes à savoir : la détection de l'application bancaire légitime et la supprimé du smartphone de la victime, la collecte d'informations saisis par la victime, la récupération des codes OTP reçu par SMS, l'envoie d'information collectées et les messages récupérés vers le compte Gmail de l'attaquant.

Une opération de test et validation de ces codes malveillants est obligatoire pour vérifier le bon fonctionnement. A l'issue de cette opération, nous allons procéder au désassemblage de l'application développée. A la fin, nous obtiendrons des fichiers .smali (l'opération de désassemblage est similaire à celle de l'étape précédente).

4.2.3 Reconstruction et signature de l'application malveillante

Dans cette étape, nous allons transformer l'application légitime en application malveillante, en injectant le code obtenu par le désassemblage de l'application légitime avec le code obtenu par le désassemblage de l'application développée. Pour cela, il suffit de copier le contenu du répertoire *smali/* du code du malware dans le répertoire *smali/* de l'application légitime.

Le code introduit ou injecté est capable de compromettre l'intégrité est capable de collecter des informations sensibles et les envoyés à distance son le consentement de l'utilisateur. Après avoir entré le code nécessaire, nous pouvons recréer l'application, mais nous devons d'abord la signer manuellement en utilisant l'utilitaire APK Signer. A la fin nous utilisons APK Studio (avec APK Tool) pour reconstruire l'application, après avoir effectué les tests de validation, l'application malveillante sera prête à être installée ou, éventuellement, à être envoyer à la victime.

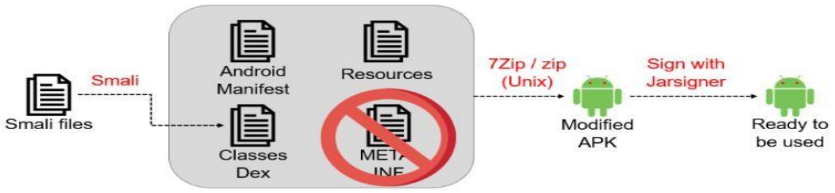


Figure 37- *Reconstruction et signature de l'application malveillante*

4.3 Diagramme de séquence de déroulement de l'attaque

Ce scénario se déclenche par l'envoi d'un lien de téléchargement d'une application malveillante (application légitime infectée par code malveillant) via une des méthodes de phishing (Mail, SMS etc..) en se passant par l'identité ou l'organisme bancaire de la victime (Usurpation d'identité). Le SMS envoyé informera la victime qu'une mise à jour de l'application bancaire est disponible et qu'il doit l'installer afin de pouvoir connecter à son compte. La victime va ensuite télécharger et installer l'application infectée, et procédera à la connexion via cette application en introduisant les informations de la carte bancaire ainsi que les informations d'authentifications (username, password), toute ces informations seront collectées par le code malveillant est envoyé à l'attaquant.

L'attaquant va ensuite procéder à effectuer des transactions frauduleuses (tel que la planification d'un retrait sans carte, ou versement vers un autres compte) et obtiendra le code à usage unique reçus par la victime par (par le service de récupération des SMS implémenté dans l'application malveillante). Cette transaction sera confirmée par l'attaquant en utilisant ce code OTP obtenu.

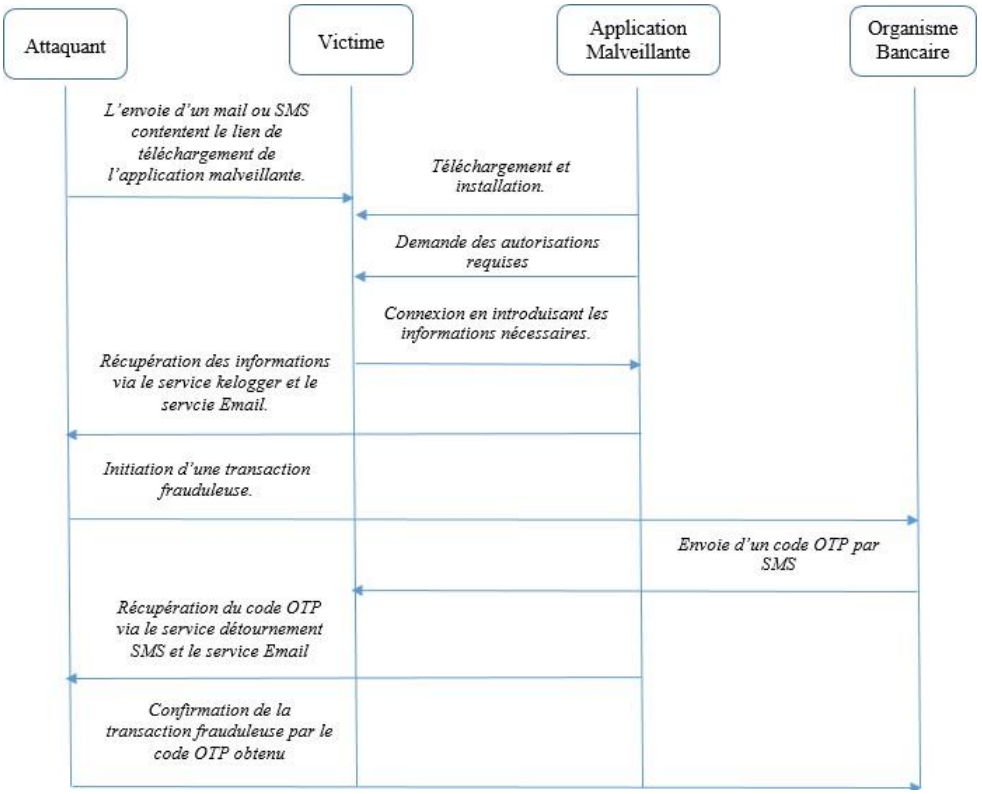


Figure 38- *Diagramme de séquence de l'attaque logiciel espion*

Conclusion

Dans ce chapitre nous avons donné une conception de plusieurs mécanismes et scénarios d'attaques qui peuvent toucher l'intégrité des cartes à puce dans le domaine bancaire et le contournement des systèmes d'authentifications mises en place, et ce, pour réaliser des actions non autorisées (*transactions frauduleuses*). Dans le chapitre suivant, nous allons mettre en œuvre ces attaques.

CHAPITRE III
Réalisation d’une attaque sur les cartes à puces (Cas de carte non-présente).

Chapitre III : Réalisation d'une attaque sur les cartes à puces

Introduction

Dans ce chapitre, nous allons présenter les outils et les étapes de réalisation et de déploiement pour chaque scénario d'attaque qui ont été cité dans le chapitre II.

1 Scénario I : Hameçonnage traditionnel

1.1 Outils utilisés

1.1.1 Le frontend

Pour le développement du Frontend nous avons choisi d'utilisé le Framework **VueJs** (la version du VueJs est 3.2.47). **VueJs** est un framework JavaScript pour la création d'interfaces utilisateur. Il s'appuie sur les standards HTML (Hyper Text Markup Language), CSS (Feuilles de Style en Cascade) et JavaScript qui aide à développer efficacement des interfaces utilisateur, qu'elles soient simples ou complexes. *[52]*

1.1.2 Le Backend

Pour le développement nous avons choisi d'utilisée le Framework ExpressJs qui est un Framework Backend gratuit et open-source. Il possède des fonctionnalités, des outils, des plugins et des paquets qui permettent de simplifier les processus de développement. Son approche minimaliste, sa grande évolutivité, sa rapidité et ses performances globales ne sont que quelques-unes des raisons pour lesquelles les entreprises et les développeurs aiment d'utiliser le Framework Express.js. Il est basé sur la Plate-forme Node.js. *[53]*

1.1.3 La base de données

Pour le stockage des données usurpées nous avons opté d'utilisée le système de gestion de base de données MongoDB à sa version 6.0.6. MongoDB est une base de données NoSQL orientée document. Elle est utilisée pour le stockage de volumes massifs de données. Contrairement à une base de données relationnelle SQL traditionnelle, MongoDB ne repose pas sur des tableaux et des colonnes. Dans la base de donnée de type NoSQL, les données sont stockées sous forme de collections et de documents. *[54]*

1.1.4 Service de messagerie

Pour que les données usurpées soient envoyés à l'attaquant par email en temps quasi-réel, nous avons choisi d'intégrer à notre site de phishing le service de messagerie **EmailJs et SmsBulk**. Ces services permettent d'envoyer des e-mails/SMS directement depuis Javascript, sans développement backend. Les développeurs peuvent créer un ou plusieurs modèles de messages (contenu dynamique pris en charge), puis ils déclenchent un message à l'aide du SDK Javascript fournis par la plateforme, en spécifiant le modèle et les paramètres dynamiques pour le rendu du message. *[55]*

1.2 Implémentation

1.2.1 Le frontend

A. Installation du NodeJs

Nous avons commencé par télécharger le binaire d'installation du NodeJs la version 18.16.0 Windows (x64) depuis https://nodejs.org/en puis nous avons procéder à l'installation.

A. Installation du VueJs et création du projet Vue

L'installation du VueJs en exécutant la commande suivante en ligne de commande :

> npm init vue@latest

La création du projet se fait par l'exécution de la commande suivante :

>npm install -g @vue/cli

B. Intégration des bibliothèques nécessaire au projet

Chaque bibliothèque est ajoutée par la commande :

>npm install [nom du pleguin]

Les bibliothèques principales utilisées sont :

- **VueRouter** : La solution recommandée par Vue pour le routage entre les composants vue;
- **Vuex** : est un gestionnaire d'état et une bibliothèque pour les applications Vuejs;
- **Axios** : est une bibliothèque JavaScript fonctionnant comme un client HTTP. Elle nous permet de communiquer avec notre API de backend en utilisant des requêtes ;
- **Emails** : Il nous permet d'envoyer des e-mails directement depuis notre code javascript.

***Figure 39**- Les bibliothèques utilisées dans notre frontend.*

c. Création des composants vue

La première étape de notre plateforme d'hameçonnage invite l'utilisateur (La victime) a introduit les informations liées à la carte bancaire ainsi que d'autres information personnelle. Ces informations constituent la première partie des données nécessaires pour passer une transaction en ligne, sachant que pour toute transaction sur un site marchand, dans l'espace de règlement de la commande, le porteur de la carte doit saisir les informations suivantes :

- Le numéro de la carte bancaire ;
- La date de validité de celle-ci ;
- Nom et Prénom du titulaire ;
- Le cryptogramme visuel (3 chiffres figurant au dos de la carte) ;
- Numéro du téléphone.

***Figure 40**- Saisie des informations de la carte bancaire.*

La deuxième étape consiste à simuler l'authentification du porteur de la carte en utilisant son téléphone portable par l'envoie d'un SMS (Code à Usage Unique *OTP*). Ce manœuvre va nous permettre d'intercepter le code à usage unique pour l'utiliser dans la validation et l'authentification d'une transaction frauduleuse.

Figure 41- *Confirmation du numéro de téléphone.*

Un code d'authentification unique (OTP) sera envoyé par l'organisme bancaire par SMS a l'utilisateur via son numéro de téléphone, ce code OTP sera intercepter par notre plateforme d'hameçonnage et utiliser dans la validation de la transaction frauduleuse.

Après l'interception du code d'authentification unique (OTP) qui a été envoyé par l'organisme à l'utilisateur, ce dernier sera utilisé pour valider la transaction frauduleuse. L'utilisateur sera redirigé, subitement, vers le vrai site web (*Original ou légal*) de l'organisme bancaire pour éviter Afin d'éviter les soupçons de la victime. Ce doit être totalement transparent et l'attaquant, dans tous les cas, aura obtenu ce qu'il voulait.

Figure 42- *envoie de la victime vers le site légitime.*

A. Architecture du frontend

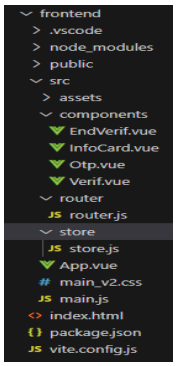


Figure 43- *Arborescence du frontend*

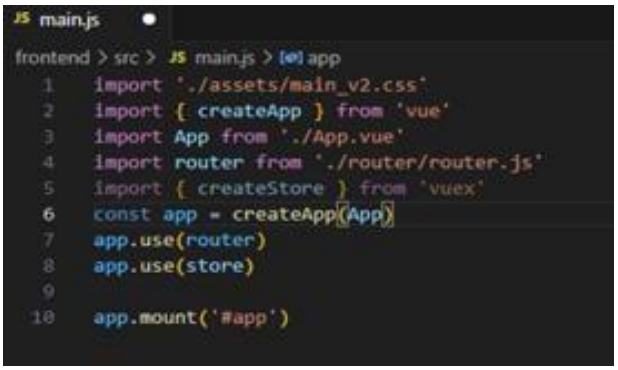


Figure 44- *Fichier main.js*

1.2.2 Le Backend

A. Initialisation du projet

A partir de notre dossier backend (le dossier qui va contenir les scripts backend), nous exécutant la commande de terminal ***npm init*** pour initialiser notre projet de backend.

> npm init

Cette commande génère un fichier package.json vierge, dans lequel seront enregistrés les détails de toutes les bibliothèques **npm** que nous utiliserons pour ce projet de backend.

B. Création du serveur

Dans cette étape, nous créons un serveur Node dans un fichier intitulées server.js qui sera le point d'entrée de notre application.

C. Création de l'application Express

Pour ajouter Express à notre projet, il suffit d'exécutez la commande :

>npm install express

L'utilisation du Framework Express simplifie les taches de développement en évitant d'analyser manuellement chaque demande entrante, en nous permettant de déployer notre API rapidement. Apres avoir installé Express, nous créons le fichier app.js qui va contenir nos fonctions, (appelées aussi **middleware)**.

```
kend > JS app.js > ...
    const express = require('express');
    const mongoose = require('mongoose');
    const Card = require('./cardModel.js');
    const Otp = require('./otpModel.js');
    const app = express();
    app.use(express.json());
    //cros policy
    app.use(function(req, res, next) {
        res.setHeader('Access-Control-Allow-Origin', '*');
        res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE');
        res.setHeader('Access-Control-Allow-Headers', 'Content-Type');
        res.setHeader('Access-Control-Allow-Credentials', true);
        next();
    });

    module.exports = app;
```

Figure 45- *Fichier App.js*

La fonction déclarer dans le code est utiliser pour que le filtre ***CORS*** permettre de passer les requêtes POST venant de notre frontend, CORS signifie « ***Cross Origin Resource Sharing*** ». Il s'agit d'un système de sécurité qui, par défaut, bloque les appels HTTP entre des serveurs différents, ce qui empêche donc les requêtes malveillantes d'accéder à des ressources sensibles.

D. Connexion avec la base de données :

Pour pouvoir connecter avec la base de données MongoDB, nous devons ajouter la bibliothèque Mongoose, cette bibliothèque nous permet de créer un schéma de données et de connecter avec la base de données MongoDB.

```
//connect to database
mongoose.connect('mongodb+srv://mahdikhaldi12:Mahdi2023$@cluster0.cht0agw.mongodb.net/cardsdb?retryWrites=true
  { useNewUrlParser: true,
    useUnifiedTopology: true })
  .then(() => console.log('Connexion à MongoDB réussie !'))
  .catch(() => console.log('Connexion à MongoDB échouée !'))
```

Figure 46- *Méthode de connexion à la base de données.*

E. Configuration des routes POST :

Nous avons créé deux routes POST, une pour sauvegarder les informations de la carte usurpées et l'autre pour sauvegarder le code OTP.

1.2.3 Installation et configuration de la base de données

Bien qu'il soit possible de télécharger et d'exécuter MongoDB sur notre propre PC, mais nous avons choisi d'utiliser la couche gratuite de MongoDB Atlas, la «database as a service» (base de données en tant que service), c'est un service de Cloud fournis par Mongo, ceci pour faciliter un future déploiement de l'application sur internet.

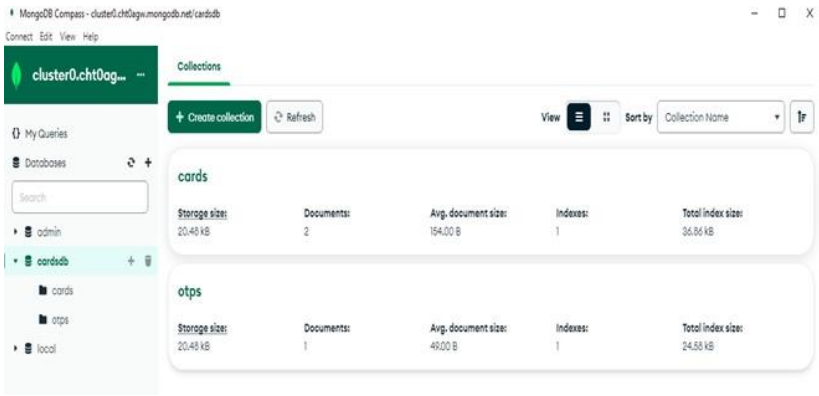


***Figure 47**- Accès à la base de données par MongoDB Compass.*

1.2.4 Création du modèle utilisée par le service de messagerie

Nous devons créer le modèle d'e-mail sur notre compte créé auparavant sur EmailJS.org, qui définira l'objet de notre e-mail, et les informations dynamiques qui serons envoyé (les données usurpées de la carte et aussi le code OTP.

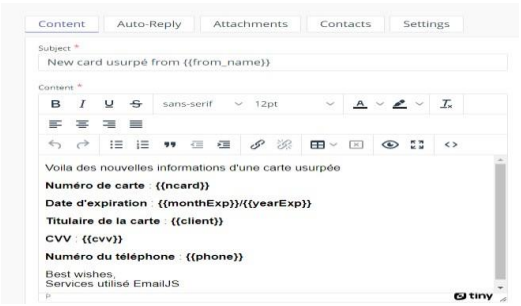


***Figure 48**- Le modèle Email*

1.3 Faisabilité

Les attaques d'hameçonnage s'appuyant sur l'ingénierie sociale se servent de la crédulité des personnes pour accéder à des données confidentielles ou les forcer à réaliser des opérations spécifiques pour obtenir un versement d'argent. Dans notre scénario qui est basé sur le phishing, on peut dire et affirmer que n'importe quel citoyen recourant à tout type de services en ligne sur Internet, notamment ceux liés aux transactions financières en lignes, est susceptible d'être une victime potentielle d'attaque de ce type (hameçonnage traditionnel).

D'après un simple scénario d'attaque basé sur l'hameçonnage traditionnel, réalisé et simulé pendant ce travail, on a constaté la faisabilité d'appliquer ce scénario sur les systèmes d'authentification en lignes mises en place par l'organisme bancaire cible qui utilise une

authentification basé sur le protocole 3DS (*Usurpation des données, réalisation et validation des transactions frauduleuses*).

On a constaté aussi, que cette manouvre peut menacer la majorité des systèmes monétiques et de paiement électronique qui utilisent les données embossées sur les cartes à puce Algérienens (Paiement en ligne et opérations bancaires en ligne). En plus, plusieurs techniques peuvent être exploitées pour automatiser le processus d'acquisition des informations et d'élargir la cible d'attaques (Grand nombre de victimes).

1.4 Mesures préventives

Comme mesures préventives de ce types d'attaques (Hameçonnage), plusieurs techniques peuvent être adoptées que ce soit au niveau de l'utilisateur final ou au niveau des systèmes de communication nationales. Ces techniques présentent une panoplie de paramètres basés sur les heuristiques, l'apprentissage automatique, les listes noires d'hameçonnage, les listes blanches et les bases de similarité visuelles.

- L'utilisation des filtres et outils anti-spam qui aident à protéger l'utilisateur des attaquants informatiques par le fait qu'ils réduisent le nombre de courriels que les utilisateurs reçoivent et par conséquent les risques d'hameçonnage.
- La mesure majeure pour prévenir ce type d'attaque reste la sensibilisation des utilisateurs, puisque ce sont les utilisateurs qui sont la cible immédiate de ces attaques, et que leur vigilance, si elle est effective, permet a priori de déjouer les tentatives d'hameçonnage.
- Privilégier l'authentification forte des utilisateurs par le renforcement des systèmes d'authentification par l'utilisation de plusieurs facteurs (Au moins deux facteurs).
- La bonne configuration des serveurs messagerie par l'intégration des différents services et méthodes mises en place comme SPF (Sender Policy Framework), DKIM (*DomainKeys Identified Mail*) et DMARC (*Domain-based Message Authentication, Reporting and Conformance*). Ces protocoles permettent d'authentifier les expéditeurs d'e-mails en vérifiant que les e-mails proviennent bien du domaine qu'ils prétendent être. Ces trois méthodes d'authentification sont importantes pour prévenir le spam, les attaques de phishing et d'autres risques liés à la sécurité des e-mails.*[62]*

2 Scénario II : Hameçonnage MITM

2.1 Outils utilisés

Pour réaliser notre reverse proxy, on s'est basé sur l'outil Modlishka ***[56].*** Modlishka est un outil open-source utilisé pour mener des attaques de phishing et de vol de session. Il fonctionne en tant que proxy inversé, permettant à l'attaquant d'intercepter les données d'authentification des utilisateurs légitimes lorsqu'ils se connectent à un site web.

2.1.1 Système d'exploitation

Le système d'exploitation sur lequel l'outil Modlishka a été installé et configuré est kali linux version 64bits. Kali Linux est une distribution Linux open source basée sur Debian, orientée vers diverses tâches de sécurité de l'information, telles que les tests d'intrusion, la recherche en sécurité, l'informatique judiciaire et l'ingénierie inverse. *[56]*

2.1.2 Go Lang

Le langage Go (ou Golang), est un langage de programmation open-source, typé statiquement. Ce langage de programmation comprend des outils qui permettent d'utiliser la mémoire en toute sécurité, de gérer les objets. Lancé en 2009 par de Google, l'objectif principal de la création de Go était de combiner les meilleures fonctionnalités des autres langages de programmation. *[57]*

2.1.3 Oracle VM VirtuelBox

Nous avons installées le système d'exploitation Kali en tant que machine virtuel en utilisant la solution de virtualisation proposé par Sun Oracle dite **Oracle VM VirtuelBox. V**irtual Box est le logiciel de virtualisation gratuit, open source et multiplateforme d'Oracle. Celui-ci permet d'héberger une ou plusieurs machines virtuelles, avec des systèmes d'exploitation différents. Grâce à ces systèmes d'exploitation invités, vous pouvez par exemple tester des logiciels sur une machine virtuelle (ou plusieurs en simultané) sans prendre le risque d'endommager votre ordinateur (hôte). *[58]*

2.1.4 OpenSSL

OpenSSL est une boîte à outils de chiffrement comportant deux bibliothèques, **libcrypto** et **libssl**, fournissant respectivement une implémentation des algorithmes cryptographiques et du protocole de communication SSL/TLS, ainsi qu'une interface en ligne de commande. *[59]*

2.1.5 Modlishka

Modlishka est un proxy inverse http. Il implémente une nouvelle approche de la gestion du flux de trafic HTTP basé sur un navigateur, qui permet de transiter de manière transparente le trafic de destination multi-domaine, à la fois TLS et non-TLS, sur un seul domaine, sans qu'il soit nécessaire d'installer un certificat supplémentaire sur le client. Ce que rend cet outil peut être utilisé dans de nombreux scénarios. *[60]*. Modlishka a été conçu comme une tentative de surmonter les limitations standard du proxy inverse et dans le but de :

- Mettre en évidence les faiblesses du schéma d'authentification à plusieurs facteurs (MFA) actuellement utilisé, afin que des solutions de sécurité adéquates puissent être créées et mises en œuvre par l'industrie.
- Sensibilisez la communauté aux techniques et stratégies modernes de phishing et soutenez les testeurs d'intrusion dans leur travail quotidien.

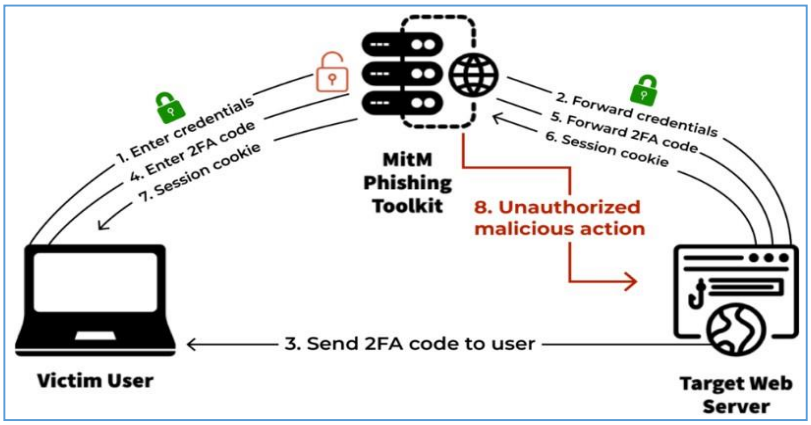


Figure 49- *Fonctionnement de Modlishka [61].*

Caractéristiques liées à la sécurité :

Voilà certaines des caractéristiques les plus importantes de Modlishka :

- Prise en charge de la majorité des schémas d'authentification 2FA (prêts à l'emploi).

- Collecte des informations d'identification de l'utilisateur (avec un contexte basé sur les identifiants passés par les paramètres d'URL).
- Plugin de panneau Web avec un résumé des informations d'identification collectées automatiquement et un module d'emprunt d'identité de session utilisateur.
- Aucun modèle de site Web (il suffit de pointer Modlishka vers le domaine cible). [*69]*

2.2 Installation et configuration

2.2.1 Préparation de l'environnement

- Téléchargement et installation d'Oracle VM Virtuel Box ;
- Installation et configuration de Kali Linux sur virtuel Box ;
- L'outil Modlishka est écrit en Golang pour cela nous avons installé Golang dans notre système Kali Linux.

2.2.2 Configuration de Modlishka

Après avoir installé Golang, nous devons définir notre GOPATH, nous allons le faire en utilisant la commande suivante :

>export GOPATH=/root/go-workspace
>export GOROOT=/usr/local/go
>PATH=$PATH: $GOROOT/bin/:$GOPATH/bin

Après avoir défini le chemin, nous devons télécharger Modlishka depuis Github pour ce faire, nous allons utiliser la commande suivante :

>go get –u https://github.com/drk1wi/Modlishka

Pour que la victime a l'impression que le site web de phishing est digne de confiance et afin d'augmenter l'efficacité de cette attaque nous devons procurons un certificat ssl, pour cela, nous utilisons l'outil openssl.

Nous générons une clé privée RSA en appliquant la commande suivante :

>openssl genrsa -out PhishingCA.key 2048

Dans la prochaine, nous allons générer un certificat SSL. La commande est la suivante :

>openssl req -x509 -new -nodes -key PhishingCA.key -sha256 -days 1024 -out PhishingCA.pem

Après avoir copié la clé, nous devons la coller dans le fichier **autocert.go** dans le répertoire du plugin en utilisons la commande **vim**.

Ensuite, nous configurons le site Web cible pour le phishing, en utilisant un fichier de configuration JSON en indiquant les paramètres principaux nécessaires :

- **Target** : contient le domaine cible ;
- **TargetRules** : Liste de chaînes statiques séparées par des virgules et leurs remplacements, tous encodés en base64 ;
- **CredParams** : Ce paramètre est utilisé par le plugin 'control panel'. Il s'agit d'une liste séparée par des virgules de modèles de correspondance des informations à collecter utilisant des groupes (regex) et encodées en base64.

Apres avoir analysée le corps http de la page web login cible, nous avons déterminé les informations à collecter et les règles d'intégrité à supprimer, puis nous l'avons mis dans les deux paramètres **targetRules** et **credParams** en base64.

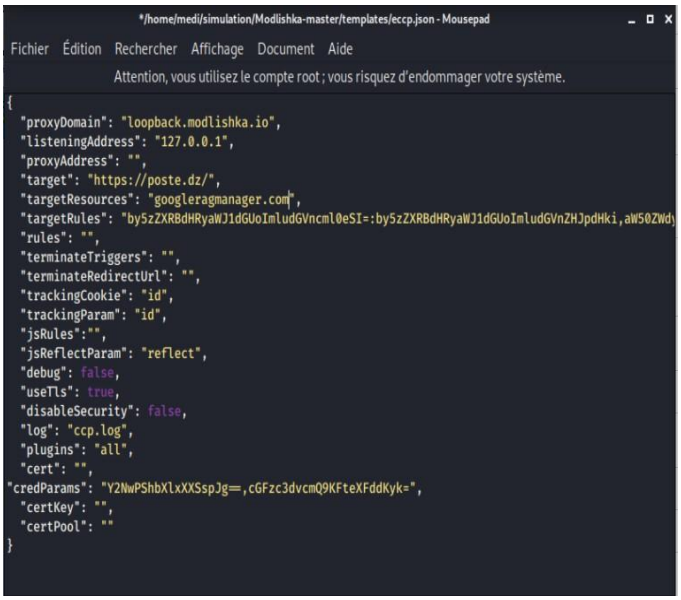


Figure 50- *Fichier de configuration JSON.*

La dernière étape pour la configuration de Modlishka consiste à compiler et lancer l'outil en utilisant les commandes suivantes:

>make

> ./dist/proxy -config templates/ccp.json

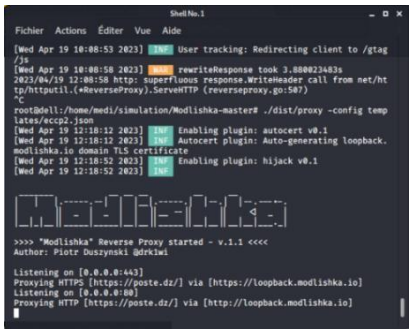


Figure 51- *Lancement de l'outil Modlishka.*

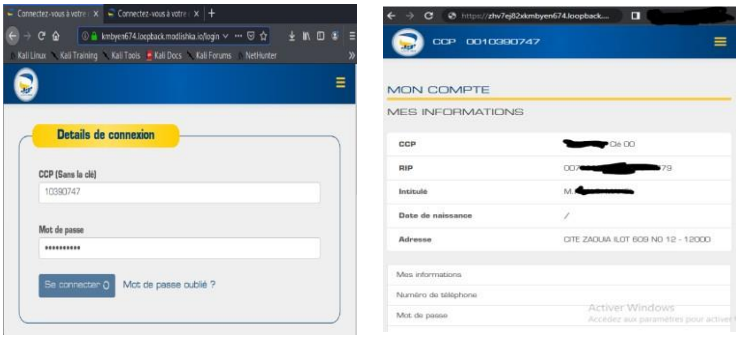


Figure 52- *Accès au site cible passant par Modlishka.*

En accédant au panneau de contrôle de Modlishka a l'url : «https://loopback.modlishka.io/SayHello2Modlsihka» ou au le fichier de journalisation nous pouvons récupérer les informations ainsi que l'identificateur de la session authentifiée.

2.3 Faisabilité :

Le domaine cible de l'organisme financier sur lequel nous avons configuré l'outil Modlishka n'utilise pas l'authentification à deux facteurs (ce qui est déjà une faiblesse), donc une attaque de type hameçonnage traditionnel suffit pour récupérer les informations de connexion au compte espace client de la victime. Cette manœuvre nous permet de changer le numéro de

téléphone enregistré dans le cadre du schéma d'authentification 3DS. L'utilisation de l'outil Modlishka dans ce scénario vise à montrer qu'il est possible de contourner les mécanismes d'authentification basés sur la 2FA (utilisés par les autres entités bancaires en Algérie). L'attaquant incite la victime à entrer ses deux facteurs dans un site de phishing qui relaie automatiquement les informations d'identification de l'utilisateur vers le site légitime en temps réel. Cette attaque rend les protections à deux facteurs inutiles car l'utilisateur croira qu'il saisit son mot de passe et son deuxième facteur sur le site légitime.

D'autre part, nous avons remarqué que les mesures de sécurité mises en place par ces organismes bancaires sur lesquels nous avons effectué des tests, telles que les claviers virtuels et le contrôle d'intégrité du DOM, sont inutiles face à Modlishka. En effet, l'architecture de type "Proxy Inverse" de Modlishka lui permet d'intercepter chaque interaction de l'utilisateur avec le site légitime et de tracer la victime tout au long de la session de navigation. Ainsi, avec une bonne configuration, cet outil est capable de contourner les règles de contrôle d'intégrité mises en place et de collecter et stocker directement les informations sensibles (informations bancaires, informations d'identification, etc.) en les extrayant du corps HTML des pages web, rendant ainsi les claviers virtuels inefficaces.

De plus, Modlishka n'est pas le seul outil disponible pour effectuer des attaques de type MITM (Man-in-the-Middle). Il existe d'autres plateformes qui aident les acteurs de phishing à transmettre les demandes de connexion aux services réels, à relayer les demandes 2FA aux victimes, puis à transmettre les codes au serveur, le tout à l'aide d'un proxy inverse. Avec des connaissances avancées dans le domaine du web, les attaquants peuvent utiliser des outils open source tels que Murène et NecroBrowser pour reproduire des attaques de contrôle de session. [63]

2.4 Mesures préventives

Du côté des utilisateurs, les mesures préventives mentionnées pour le scénario d'hameçonnage traditionnel sont également valables pour atténuer l'hameçonnage de type MITM. Du côté des sites légitimes, les serveurs utilisés pour l'authentification des utilisateurs pourraient identifier et filtrer les demandes de transfert malveillantes en utilisant uniquement une clé matérielle, qui est considérée comme le facteur de possession le plus efficace. Ces clés matérielles utilisent un protocole appelé U2F, et plus précisément WebAuth, pour communiquer avec le site légitime. Ces protocoles peuvent atténuer ce type d'attaques en effectuant des vérifications pour s'assurer que le site cible est bien le site légitime en utilisant la cryptographie à clé publique [64]. Cependant, l'industrie en est encore loin à l'heure actuelle.

Actuellement, la seule défense largement utilisée contre ces attaques consiste à utiliser des listes de blocage basées sur les adresses IP, actualisées quotidiennement. Cependant, cela n'est évidemment pas suffisant et l'industrie doit continuer à développer des mesures de sécurité plus avancées pour contrer efficacement les attaques de ce type. [65]

3 Scénario III : Logiciel espion

3.1 Outils utilisés

Cette approche repose sur l'utilisation d'apk falsifié. On a privilégié l'utilisation d'ApkTool et Apk Studio.

3.1.1 APK Studio

Pour le désassemblage et la reconstruction des applications nous avons travaillé avec APK Studio (la version 5.2.4-x64), qui est un outil puissant de rétro-ingénierie. APK Studio est un outil open source disponible pour Windows, MacOs ainsi que Linux, utilisé pour la création des applications Android, mais il peut être utilisé aussi pour décompiler et recompiler les fichiers Android APK, modifier le code d'une application existante ou debugger une application *[66]*.

3.1.2 APK Tool

Nous avons paramétré APK Studio pour utiliser ApkTool la version 2.6.0. ApkTool est un fichier au format JAR considérer aussi comme un outil rétro-ingénierie. Il peut décoder les ressources sous une forme presque originale et les reconstruire après avoir apporté quelques modifications ; il permet de déboguer le code smali étape par étape *[67]*. Cela facilite également le travail avec l'application en raison de la structure des fichiers de type projet et de l'automatisation de certaines tâches répétitives telles que la création d'apk *[68]*.

3.1.3 SDK Platform Tools

SDK Tools est nécessaire pour le fonctionnement d'APK Studio, est un composant du SDK Android. Il comprend des outils qui offrent une interface avec la plate-forme Android, principalement adb et fastboot. Bien qu'adb soit nécessaire au développement d'applications Android *[69]*.

3.1.4 Uber APK Signer

Nous avons paramétré APK Studio pour utiliser Uber-APK-Signer la version 1.3.0, il permet de signer des APK et de confirmer que la signature d'un APK sera vérifiée avec succès sur toutes les versions de la plate-forme Android prises en charge par cet APK *[70]*.

3.1.5 Android Studio

Pour le développement de l'application Android basique contenant les codes malveillants nous avons travaillé avec Android Studio. Il est l'environnement de développement intégré (IDE) officiel des applications Android. Basé sur le puissant outil de développement et d'édition de code d'IntelliJ IDEA *[71]*. Il est considéré comme un environnement unifié pour le développement sur tous les appareils Android, grâce à ses fonctionnalités qui améliorent la création des applications Android. *[72]*

3.2 Implémentation

3.2.1 Désassemblage de l'application légitime

Apres avoir effectué le paramétrage nécessaire pour le fonctionnement de APK Studio (l'ajout des fichiers JAR APK Tool, Uber-APK-Signer et l'exécutable ADB Tool de SDK Platform Tools), nous avons procéder au désassemblage de l'application bancaire légitime en utilisant APK Studio (Voir figure 53). Parmi les répertoires et les fichiers produits on distinct répertoire smali qui contient Les fichiers .smali reflètent le comportement de l'application d'origine, nous avons également les répertoires original et unknown :

- 'original' contient le dossier META-INF et le fichier xml nécessaires lorsque nousvoulons

reconstruire l'application tout en conservant la signature d'origine ;

- 'unknown' contient des fichiers et des dossiers qui ne font pas partie du projet Open Source Android (AOSP).
- Le fichier AndroidManifest contient l'implémentation des services fournis par l'application légitime ; *[73]*

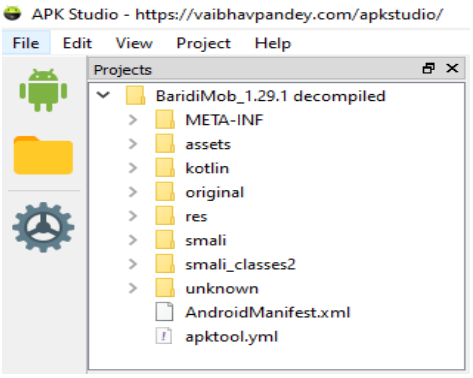


Figure 53- *Répertoire et fichier produits par le désassemblage de l'application légitime.*

3.2.2 Préparation du logiciel malveillant

Dans cette étape, nous utilisons Android Studio pour développer une application Android basique contenant plusieurs codes malveillants. Nous avons commencé par la création d'un nouveau projet d'une application vierge puis nous avons procéder au développement des fonctionnalités malveillantes.

Les services que nous allons créer ce sont des composants d'arrière-plan qui peut effectuer des opérations en arrière-plan. Il ne fournit pas d'interface utilisateur, pour déclarer le service il suffit d'ajouter un <service> élément en tant qu'enfant de l' <application> élément.

Pour chaque service créé nous devons déterminer les autorisations nécessaires pour l'exécution de ces services et les déclarer dans le fichier **AndroidManifest.** *[74]*

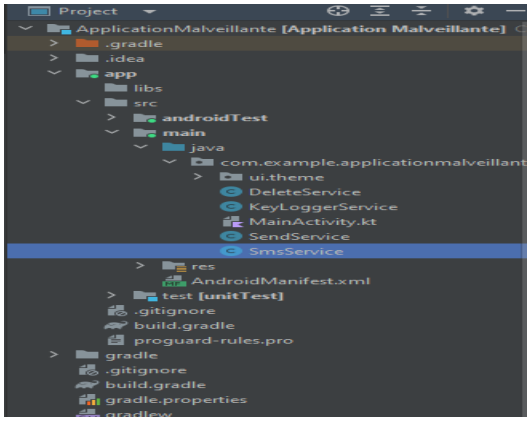


Figure 54- *Structure de l'application malveillante.*

Figure 55- *Déclaration des services dans le fichier AndroidManifest.xml.*

Figure 56*- Déclaration des autorisations dans le fichier AndroidManifest.xml.*

Voilà une brève description de chaque service :

- **DeleteService**

Ce service est chargé de la détection et de la suppression de l'application bancaire légitime, l'implémentation de ce service utilise l'action : « ACTION_UNINSTALL_PACKAGE » qui hérite de la classe « Intent» *[75]*

- **KeyLoggerService**

C'est le service de collecte des informations, il sera utilisée pour recueillir des informations sur les frappes sur les téléphones mobiles des victimes, un tel programme est appelé Keylogger. Il surveille et conserve des enregistrements de chaque frappe enregistrée sur un clavier dans des fichiers journaux spéciaux à l'insu de l'utilisateur de l'appareil *[76]*. Nous avons essayé d'implémenter ce service en héritant de la classe KeyEvent *[77]*.

Mais dans les dernières versions Android et pour des mesures de sécurités, la journalisation des touches pour les événements de clavier n'est pas prise en charge dans les services d'arrière-plan. Il nous reste une autre piste à explorer implémenter ce service, est en utilisant les API d'accessibilité *[78]*, mais cette méthodes nécessite des compétences avancées en terme de développement des applications Android. Cette méthode est déjà utilisée pour implémenter des applications KeyLogger connues sur le marché comme mSpy, eyeZy et Snoopza [*76*].

- **SmsService**

Service de récupération des SMS, il utilise la classe «Telephony.Sms» [79]. Le rôle de ce service est de parcourir le corps des messages rentrants et récupérer ceux provenant de l'organisme bancaire émetteur de l'application légitime (des sms qui contiennent des codes OTP, des codes pin d'une opération de retrait sans carte etc.).

- **SendService**

Service d'envoi des informations collectées et des SMS récupérer. Le rôle est d'envoyé les informations collectées et les SMS récupérés par Mail vers l'attaquant. Ce service va être implémenté à l'aide de l'API Java Mail et les dépendances (mail.jar, activation.jar, additionnal.jar) [80].

3.2.3 Reconstruction et signature de l'application malveillante

Cette étape se résume en :

- Injection du code obtenu par le désassemblage de l'application légitime avec le code obtenu par le désassemblage de l'application développée. Pour cela, il suffit de copier le contenu du répertoire *smali/* du code du malware dans le répertoire *smali/* de l'application légitime ;

- Reconstruction de l'application par l'outil APK Studio ;
- signature de l'application en utilisant APK Studio (via la bibliothéque Uber-APK Signer).

3.3 Faisabilité

L'ingénierie sociale, dans notre scénario, implique l'envoi d'une application malveillante à la cible via un SMS. Cette application sera installée sur le smartphone de la victime et permettra le vol de données sensibles ainsi que le contournement de l'authentification à deux facteurs (2FA).

Un scénario d'attaque basé sur la transformation d'une application légitime d'un organisme bancaire en une application malveillante constitue une menace potentielle en Algérie, notamment en ce qui concerne l'utilisation d'applications financières telles que l'application BaridiMob d'Algérie Poste. La transformation d'une telle application en un code malveillant nous permet de voler des informations relatives à la carte bancaire ainsi que les informations de connexion utilisées pour cette application bancaire (nom d'utilisateur, mot de passe). Cette manœuvre offre également la possibilité de contourner l'authentification à deux facteurs (2FA).

3.4 Mesures préventives

Il y en a pas mal de logiciels malveillants ciblant les systèmes Android. En effet, ces malwares sont capables d'exécuter des fonctions d'espionnage, notamment collecter les données sensibles des victimes, intercepter les SMS, enregistrer les frappes au clavier, installer/ désinstaller des applications etc. Comme mesures préventifs contre ce type d'attaques, plusieurs techniques peuvent être adoptées au niveau de l'utilisateur final tel que [81]:

- Maintenir la mise à jour automatique de vos systèmes Android ;
- Eviter d'installer des applications depuis des sources non fiables et non officielles ;
- limiter au minimum nécessaire le nombre d'applications installées sur votre appareil et à veiller à ce que les autorisations demandées ne soient pas trop larges ;
- Scanner l'appareil immédiatement par un anti-virus mis à jour aprèsl'installation de chaque nouvelle application.
- Installer une solution anti-spam pour contrer les messages suspects mobiles ;

Conclusion

Dans ce chapitre nous avons procédé à l'implémentation des scénarios d'attaques qui peuvent toucher l'intégrité des cartes à puce dans le domaine bancaire et le contournement des systèmes d'authentifications mises en place, et ce, pour réaliser des actions non autorisées (*transactions frauduleuses*).

Conclusion général et Perspective

Assurer la sécurité dans le domaine des cartes à puce est une tâche très importante en raison de la sensibilité des données qu'elles contiennent (données bancaires et financières, codes d'accès aux comptes, informations personnelles, données biométriques, données médicales, etc.). Malgré les mécanismes et les techniques déployés depuis l'avènement de ces cartes, nous avons constaté dans ce travail que plusieurs menaces ont ciblé l'intégrité de ces cartes, que ce soit en leur présence physique ou lors des transactions à distance, comme les paiements en ligne.

Nous avons étudié dans ce mémoire les cartes à puce et les différentes attaques qui ont ciblé ces cartes, en mettant l'accent sur leur utilisation dans le domaine des transactions bancaires. Cela est dû à l'adoption, ces dernières années, de plusieurs solutions monétiques électroniques basées sur l'utilisation de cartes à puce en Algérie, notamment la carte EDAHABIA d'Algérie Poste et la carte CIB émise par les différentes banques algériennes et étrangères présentes sur le territoire national.

Pour atteindre nos objectifs, nous avons :

- Exploré le domaine des cartes à puce, en particulier celles embarquant une plate-forme de type Java Card, car elles représentent la majorité des cartes en circulation dans le monde. Cette partie a également examiné les mécanismes de sécurité mis en place pour garantir la sécurité des transactions, qu'elles soient de proximité ou à distance.

- Identifié et compris les différentes typologies d'attaques et de menaces qui ont ciblé les cartes à puce, notamment dans le domaine monétique. Ces techniques ont réellement compromis l'intégrité de ces cartes, que ce soit physiquement en s'attaquant aux composants matériels, logiquement en exploitant les couches applicatives et système, ou encore lors des transactions à distance lors de la transmission des données en ligne.

- Enfin, nous avons mis en place une simulation d'attaque sous forme de scénarios, ciblant le vol et l'utilisation frauduleuse des données d'une carte bancaire (de type carte à puce) lors d'une transaction en ligne. Dans notre cas, la simulation a ciblé le système d'authentification des transactions en ligne, en particulier le protocole 3D Secure (3DS) utilisé par le système de paiement en ligne en Algérie.

À partir de nos recherches, nous avons conclu qu'actuellement, la majorité des attaques et des fraudes ciblant les cartes bancaires (cartes à puce) se produisent en ligne, en l'absence physique de la carte elle-même. Cela est dû à la mise en place de nombreuses mesures de sécurité au niveau de la carte elle-même, tant au niveau matériel que logiciel, qui ont permis de prévenir la majorité des attaques physiques ou logiques. Par conséquent, dans notre simulation, nous avons choisi de cibler les données d'identification et d'authentification lors d'une transaction en ligne (carte non présente).

Le système de paiement électronique en Algérie s’articule principalement sur l’utilisation de la carte CIB émise par les différentes banques et la carte EDAHABIA d’Algérie poste (*environ 15 millions cartes*), et ce, à travers tout un réseau interconnecté grâce aux solutions monétiques électroniques mises en place par la Sociétéd'Automatisation des Transactions Interbancaires et de Monétique (*SATIM*), Algérie Poste, le Groupement d’Intérêt Economique Monétique (*GIE-Monétique*), la Banque d’Algérie (*BA*).

Actuellement, le système de paiement électronique en Algérie se base sur l’utilisation du protocole 3DS pour sécuriser les transactions en ligne et s’assurer la bonne authentification du porteur de la carte. Ce protocole sert, entre autres, de garantir que c’est bien celui-ci qui effectue

la transaction sur internet et n'est pas une personne malveillante.

Nous rappelons que ce travail est purement pédagogique. A travers ce dernier, nous avons montré les limitesde l'utilisation du protocole 3DS en Algérie en termes de sécurité. Il n'y a pas de réelles perspectives, mise à part la recommandation aux autorités concernées d'étudier le passage à la 3DS2. Ce dernier se base essentiellement sur le stockage et l'exploitation des données à caractère personnel. (Contraintes juridiques et techniques, en vue que les modalités de stockage et de sécurisation desdonnées à caractère personnel sont définies conformément à la législation et à la réglementationen vigueur, notamment la loi, notamment la loi n° 18-07 du 25 10 juin 2018 relative à la protectiondes personnes physiques dans le traitement des données à caractère personnel).

Bibliographie

[01] ***Nilson Reports 2022, 2021*** *; la source d'actualités et d'analyses la plus respectée de l'industrie mondiale des paiements par carte bancaire.*

[02] ***Julien Iguchi-Cartigny ;*** *Contributions à la sécurité des Java Card ; Mémoire HDR. Université de Limoges, Décembre 2014.*

[03] ***J.L.Lanet, Pascal Urien*** *; Support de cours : Carte à puce ; 2008, 2010,2020.*

[04] ***Dichou Karima****, Contribution à l'étude des cartes à puce avancées, Thèse de Doctorat, 2016.*

[05] ***Hamadouche Samiya ;*** *Étude de la sécurité d'un vérifieur de bytecode et génération de tests de vulnérabilité, Thèse de Magistère, juin 2012.*

[06] ***Pascal Chour*** *; cours carte à puce ; Historique, concepts, sécurité des cartes. Version 2021.*

[07] ***Boumassata Meriem*** *; Vérification de code pour plates-formes embarqués, Thèse de fin d'étude d'un Magistère, Université de Batna, 2012.*

[08] ***Damien Sauveron*** *; Étude et réalisation d'un environnement d'expérimentation et de modélisation pour la technologie Java Card. Application à la sécurité. thèse doctorat décembre 2004.*

[09] ***Sun Mycrosystems*** *; Java Card Data Sheet ; 2019.*

[10] ***Wolfgang Rankl and Wolfgang Effing****, Smart Card Handbook (Book), Giesecke & Dwievrient GmbH, Munich, Germany.*

[11] *Sun Mycrosystems ; Java card development kit user guide ; Version 3.1.0u5 ; E99052 07 March 2021.*

[12] ***Samiya Hamadouche, Jean-Louis Lanet2, Mohamed Mezghiche1 ;*** *Méthode d'Analyse de Vulnérabilité Appliquée à un Composant de Sécurité d'une Carte à Puce. In Rencontres sur la Recherche en Informatique ; Tizi-Ouzou, Algérie, Juin 2011.*

[13] ***Xavier Kauffmann-Tourkestansky*** *; Analyses sécuritaires de code de carte à puce sous attaques physiques simulées, Université d'Orléans, Novembre 2012.*

[14] ***C.BARRAL;*** *Biometrics & Security: Combining Fingerprints, Smart Cards and Cryptography. Thèse de doctorat, Faculté Informatique Et Communications. Laboratoire De Sécurité et De Cryptographie, Suisse, 25 Août 2010.*

[15] ***A. Karra ;*** *Conception, mise en oeuvre et validation d'un environnement logiciel pour le calcul sécurisé sur une grille de cartes à puce de type Java. Thèse de doctorat, Université Bordeaux I, 10 décembre 2008.*

[16] *W****. Rankl, W. Effing****, Smart Card Handbook, 4th ed, John Wiley & Sons, New York, 2010.*

[17] ***K. Markantonakis and K.E. Mayes****, Smart Cards, Tokens, Security and Applications. Springer Science & Business Media, University of London, 2007.*

[18] ***Baptiste Besson, Magid Aberkane*** *; Les Java Cards ; rapport technique ; Université de Nice-Sophia Antipolis ; Juin 2004.*

[19] ***Raja Naeem Akram*** *; Trusted Platform Module for Smart Cards ; Cyber Security Lab, Department of Computer Science University of Waikato, Hamilton. New Zélande.*

[20] ***Sergei Volokitin****, Good, Bad and Ugly Design of Java Card Security, Radboud University, thèse de Mastère ; June 2016.*

[21] ***Sun Microsystems;*** *The Java Card™3 Platforme, Août 2008. White Paper.*

[22] *http://www.satim.dz : Site officiel de la société SATIM ; Société d'Automatisation des Transactions Interbancaires et de Monétique.*

[23] ***Germain Jolly.*** *Evaluation d'applications de paiement sur carte à puce. Cryptographie et sécurité [cs.CR]. Normandie Université, France, 2016.*

[24] ***Nicolas Guay****, Sécurité des cartes de paiement, Université de Montréal mars 2011.*

[25] ***Directive DSP2*** *(UE) 2015/2366 Du Parlement Européen Et Du Conseil, du 25 novembre 2015.*

[26] ***European EMV 3DS 2.2.0*** *Implementation Guide October 2019.*

[27] *https://www.emvco.com; Site officielle de l'organisme EMVCO qui s'occupe du standard EMV.*

[28] ***W. Rankl, W. Effing,*** *Smart Card Handbook, 4th ed, John Wiley & Sons, New York,2010.*

[29] ***Ahmadou A.Séré****: Tissage de contremesures pour machines virtuelles embarquées. Thèse de doctorat, Université de Limoge, 2010.*

[30] ***MAQUA Fabrice,*** *Cartes de paiement et contrefaçon développement d'un outil d'analyse de cartes de paiement, Thèse de fin d'étude, université de technologie de Troyes, 2014.*

[31] ***Observatoire de la sécurité des moyens de paiement****, Rapport annuel 2020, Rapportannuel 2019, Rapport annuel 2018 .*

[32] ***BB Gupta et Megha Quamara*** *; Une taxonomie de diverses attaques sur des applications basées sur des cartes à puce et des contre-mesures, Institut national de technologie Kurukshetra, Kurukshetra, Inde, 2018.*

[33] ***Marc.Witteman :*** *Advances in smartcard security ; Information Security Bulletin, (July): 2022.*

[34] ***C.Aumüller, P.Bier, W.Fischer, P.Hofreiteret J.P.Seifert;*** *Fault attacks on RSA with CRT: Concrete results and practical countermeasures ; Scientific Article.*

[35] ***Michel Agoyan, Jean-Max Dutertre, David Naccache, Bruno Robisson, and Assia Tria*** *; When Clocks Fail : On Critical Paths and Clock Faults, 2010.*

[36] ***Sergei P. Skorobogatov and Ross J. Anderson*** *; Optical fault induction attacks; University of Cambridge,2003.*

[37] ***Jörn-Marc Schmidt et Michael Hutter****; Optical and EM Fault-Attacks on CRT-based RSA: Concrete Results , Graz University of Technology , Austria, 2017.*

[38] ***Ibrahima DIOP ;*** *Méthodologie et outils pour la mise en pratique des attaques par collision et attaques horizontales sur l'exponentiation modulaire ; thèse de doctorat de l'université de Lyon ; 2017.*

[39] ***Lilian Bossuet;*** *Approche didactique pour l'enseignement de l'attaque DPA ciblant l'algorithme de chiffrement AES. Journal sur l'enseignement des sciences et technologies de l'information et des systèmes, EDP Sciences, 2012.*

[40] ***Barbu G, Duc G, Hoogvorst P:*** *Java Card operand stack: fault attacks, combined attacks and countermeasures. In: Smart Card Research and Advanced Applications (CARDIS); 2011.*

[41] ***Sergei Volokitin and Erik Poll ;*** *Logical attacks on secured containers of the Java Card platform. Digital Security group, Radboud University Nijmegen, The Netherlands, 2016.*

[42] ***J. Hogenboom and W. Mostowski;*** *Full memory read attack on a java card. In Proceedings of Workshop on Information and System Security (WISSEC), 2009.*

[43] ***G. Dalton, R. Mills, J. Colombi, and R. Raines****, "Analyzing Attack Trees using Generalized Stochastic Petri Nets," 2006 IEEE Information Assurance Workshop, 2006, pp. 116-123.*

[44] ***British Broadcasting Corporation :*** *Trojan virus steals banking; http://news.bbc.co.uk/2/hi/technology/7701227.stm, octobre 2008.*

[45] ***Mohamed. Z,*** *La sécurité de la plateforme java card, thèse de Mastère, Ecole Supérieur polytechniques, Juin 2018.*

[46] ***Souvignet Tomas ;*** *l'expertise et la lutte contre la fraude monétique ; Thèse doctorat ; Université Panthéon ; Décembre 2014.*

[47] ***Eoropean Central Bank :*** *Seventh report on card fraud October 2021****.***

[48] ***Francisco Corelle Karen Lewisson ;*** *Fundamental Security Flaws in the 3-D Secure 2 Cardholder Authentication Specification; Octobre 2019.*

[48] ***Global Fraud and payements Report ;*** *Cybersource et le Merchant Risk Council 2022****.***

[49] *https://github.com/drk1wi/Modlishka, Site officiel github.*

[50] *stats Counter GlobalStats :mobile ,Operating System Market Share Worldwide, Apr 2022 - Apr 2023, Disponible : https://gs.statcounter.com/os-market-share/mobile/worldwide*

[51] *Rapport sur les statistiques et les tendences des logiciels malveillant consulté le 15 Mai 2023 Disponible : https://www.av-test.org/de/statistiken/malware.*

Webographie

[52] *https://vuejs.org.*

[53] *https://expressjs.com.*

[54] *https://www.mongodb.com.*

[55] *https://www.emailjs.com.*

[56] *https://www.kali.org/docs/introduction/what-is-kali-linux*

[57] *https://go.dev/doc/*

[58] *https://www.oracle.com/virtualization/virtualbox*

[59] *https://fr.wikipedia.org/wiki/OpenSSL*

[60] *https://fr.wikipedia.org/wiki/Proxy_inverse*

[61] *https://github.com/drk1wi/Modlishka*

[62] *https://www.cloudflare.com/fr-fr/learning/email-security/*

[63] *https://github.com/muraenateam/necrobrowser*

[64] *https://fr.wikipedia.org/wiki/WebAuthn*

[65] *https://fr.wikipedia.org/wiki/Universal_Second_Factor*

[66] *https://androidmtk.com/apk-studio*

[67] *https://www.file-extension.org/fr/extensions/smali*

[68] *https://github.com/vaibhavpandeyvpz/apkstudio*

[69] *https://developer.android.com/studio/releases/platform*-tools

[70] *https://github.com/patrickfav/uber-apk-signer/*

[71] *https://www.jetbrains.com/idea/*

[72] *http://developer.android.com/studio*

[73] *https://developer.android.com/studio/projects*

[74] *https://developer.android.com/training/permissions/*

[75] *https://developer.android.com/reference/android/content/Intent*

[76] *https://www.javatpoint.com/best-keylogger-for-android*

[77] *https://developer.android.com/reference/android/view/KeyEvent*

[78] *https://developer.android.com/guide/topics/ui/accessibility*

[79] *https://developer.android.com/reference/android/provider/Telephony.Sms*

[80] *https://www.oracle.com/java/technologies/javamail.html*

[81] *https://www.kaspersky.fr/resource-center/preemptive-safety/avoid-android-malware*