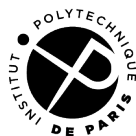


Ceci est ma thèse

Vie amoureuse des tétards à Ph7

Maxime Ayrault

A thesis presented for the degree of
Doctor of Philosophy



IP PARIS

Department Name

Telecom Paris

France

30 février 2056

Ceci est ma thèse

Vie amoureuse des têtards à Ph7

Maxime Ayrault

Résumé

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Dédicace

À ma Maman adorée...

Remerciements

Je remercie Jean-Do pour ces conseils avisés.....

Table des matières

1	Introduction	7
2	Security in the automotive world	8
2.1	The software revolution	8
2.2	Modern car software architecture	9
2.2.1	AUTOSAR	9
2.2.2	The software stacks	11
2.3	Cyber-security principles	13
2.4	Le risque cybersécurité pour l'automobile	16
3	La théorie des jeux	18
3.1	Présentation	18
3.2	Définition générale	19
3.2.1	Types de jeux	19
3.3	Stratégie pure	20
3.3.1	Dominance	20
3.3.2	Équilibre	21
3.4	Stratégie mixte	21
4	Expression de la résilience	22
4.1	Problématique	22
4.2	Changement d'adressage dynamique	22
4.3	La théorie des jeux	22
4.4	Apprentissage	22
5	Conclusions et perspectives	23
5.1	Les leçons apprises	23
5.2	Les frontières de la recherche	23
A	Ma belle annexe	27
A.1	Introduction	27

Table des figures

2.1	AUTOSAR layered architecture	11
2.2	Architecture générique	13
2.3	3 pillars of the security	14

Liste des tableaux

2.1	Software function in a car	8
2.2	Principales méthodes de la sécurité informatique	16
4.1	very basic table	22

Chapitre 1

Introduction

Il est très important, pour celui qui souhaite découvrir, de ne pas limiter son esprit à un seul chapitre de la science mais plutôt de rester en contact avec plusieurs autres.

Jacques Hadamard.

Nous vivons dans l'ère des télécommunications. Les communications constituent, sans aucun doute, l'une des plus importantes révolutions de la science et la technologie.

Ce mémoire de thèse

Ce mémoire est divisé en six chapitres et 2 annexes..

Le chapitre 2 contient un rappel sur la théorie....

Le chapitre 3 développe....

Au chapitre 4, expose une application des résultats

Le chapitre 5 déjà la conclusion et les perspectives

Dans l'espoir de pouvoir intéresser le lecteur curieux,

Chapitre 2

Security in the automotive world

This work was presented in part at the conference of one-legged deaf-mutes in Quiberon in April 1994.

Remoi

2.1 The software revolution

Embedded software is one of the key innovation in the automotive world. From the paper of Robert Charette (CHARETTE 2009) published in IEEE Spectrum, the first car embedding a software was the Oldsmobile Toronado from General Motors in 1977. The Toronado enclosed an Electronic Control Unit (ECU) which managed the spark timing. In 1978, General Motors offered on the Cadillac Seville as an option a trip computer able to display the speed, the fuel level, trip and engine information. This product was based on an embedded version of a Motorola 6802 and had about 50,000 lines of code. Since, more and more functions are performed by software in a car. To limit the volume of cable in a car, all the sensors and on-board unit are now connected to a network backbone using CAN or xxx network. As the computing power of the processor grows, new functions appear in cars. Cars become now on-wheel software platform like airplanes or trains. A modern family car embeds between 30 and 50 ECUs performing the management of the multiple systems(see 2.1). A premium car can have up to 3,000 singular functions performed by software.

Airbag	ABS	Anti-thief system
Air Conditioning	Speed control	motor management
Turn signals	Headlights	Klaxon
Seat management	Navigation system	Audio system
Wheel pressure	management of the doors and windows	...

TABLE 2.1 – Software function in a car

In 2009, Alfred Katzenbach, Director of Information Technology Management at Daimler announced that the radio and navigation management system of an S-class Mercedes-Benz contains over 20 millions line of code and the car embeds nearly as many ECUs as an Airbus A380 (excluding the in-flight entertainment system) (CHARETTE 2009). Software in a car has an exponential growth in size and complexity. In just 10 years, the software volume in a car expands by a factor 10, to arrive at around 150 million of lines of code. A model S from Tesla is equipped with a 17' tactical display based on a Linux kernel which controls almost every driver functions. In fact, there is only 2 manual buttons taht are not management by software in the car, the blinker and the glove's compartment.

A side effect of this revolution is the complexity and the richness of the functions proposed to the driver. It is frequent to have the driver manual of a car with over 500 pages to explain all the driver functions. Automotive experts estimate that an average driver uses not more than 20% of those functions. Thus is why, some automotive manufacturers are asking them the question of a limitation (or reduction) of software functions proposed in a car. For example, is it really necessary to have the ceiling light of a car going out gradually when the doors are closing ?

The growth of the software in a car has also important consequences on the way to maintain and repair a car. An estimation gives that more than 50% of the ECU that are changed by a car mechanic have no software or hardware failure. Mechanic frequently replaces pieces without the knowledge of the root cause of the issue. Now, one of the most frequent activity of a car mechanic is to download and upgrade new version of software. Some car manufacturer, like Tesla proposes to download software updates, including software corrections (patches) or new functionalities, using the cellular network without any involvement of a car mechanic.

Today, the cost of the software and the supporting electronic in a car is estimated between 35% and 40% of the cost of a car. Investment to develop new software platform became so expensive that the main European Automotive manufacturers develop a series of standards to reduce the developent cost :

- a commom platform ; *Automotive Open System Architecture (AUTOSAR)* AUTOSAR 2003 allowing there suppliers to develop a single platform inter-operable between several manufacturers.
- a software development framework for the safety ; *Road vehicles, Functional Safety package* ISO 26262 STANDARDIZATION 2018
- and recently a standard to address the cybersecurity management system ; *Road vehicles - Cybersecurity engineering* ISO 21434 STANDARDIZATION 2021.

2.2 Modern car software architecture

2.2.1 AUTOSAR

AUTomotive Open System Architecture (AUTOSAR) was founded in 2003, with the goal to develop an architecture, independent of the underlying ECU hardware that the automotive industry can use to reduce the increasing complexity of software in modern vehicles (AUTOSAR 2003). This is the de facto standard for the automotive software today. AUTOSAR makes an abstract layer of the underlying hardware, so that the applications written on top of AUTOSAR are independent from the ac-

tual supplier of the ECU hardware. The AUTOSAR standard documentation guides companies and the automotive industry in designing and implementing software in their vehicles. Besides the Software architecture and the Standardized application interfaces (APIs), AUTOSAR provides a Software development methodology. By adopting the AUTOSAR standard, companies can develop software solutions that are independent of the hardware they are running on, and this software can run on any ECU in the vehicle. Threat and Application Programming Interface (API)

AUTOSAR is a three-layered architecture (AUTOSAR 2020) :

- the *application layer* provided by the software company implementing the specific functions of the ECU. This is the highest layer which contains the software components (SWCs). AUTOSAR application (e.g., ABS, Cruise Control ...) consists of several SWCs, which provide the core functions. An AUTOSAR SWC is an atomic piece of software that cannot be divided and is located a single ECU.
- the *run-time environment (RTE) layer*. The RTE layer provides the standardized interface between the SWCs and the basic software layer. Because of this layer, SWCs can be used on different ECUs, independent of the ECU vendor.
- the *basic software (BSW) layer* that consists of four sub-layers; the *services layer* providing operating system functions like communication services, memory services, diagnostic services, The layer also contains the main security mechanisms. The *ECU abstraction layer* makes higher software layers independent of ECU hardware layout. It provides an application programming interface to devices regardless of their location (internal/external of the microcontroller). The *Microcontroller abstraction layer*. It contains drivers for direct access to the underlying microcontroller and internal parameters. It makes higher layers independent of the microcontroller. The *complex drivers layer* which provides the ability to integrate special-purpose functions such as drivers for devices that are not specified with the AUTOSAR standard. This layer accesses directly the microcontroller.

Each of the sublayers offers different services as shown in 2.1.

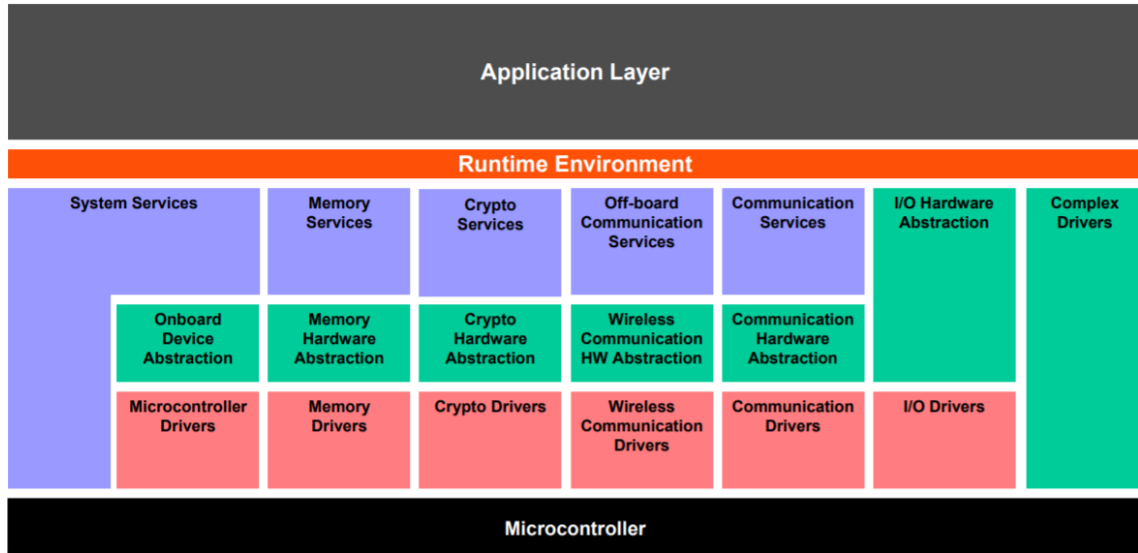


FIGURE 2.1 – AUTOSAR layered architecture

The AUTOSAR standard defines security mechanisms that can be used by the software modules implemented into the vehicle system. It further specifies interfaces and procedures to provide Secure On-Board Communication, and the exact implementation is left for the OEMs to decide on. OEMs choose the cryptographic algorithms and encryption techniques which they want to implement and use in the vehicle system.

Introduction of software in the car permits to have safer and less polluting cars but it has open the door at two new risks the *safety risk* and the *cyber-security threat*.

2.2.2 The software stacks

Todo list 1. *Commencer a présenter les différentes couches logiciel.*

The following four stacks could become the basis for upcoming generations of cars in five to ten years :

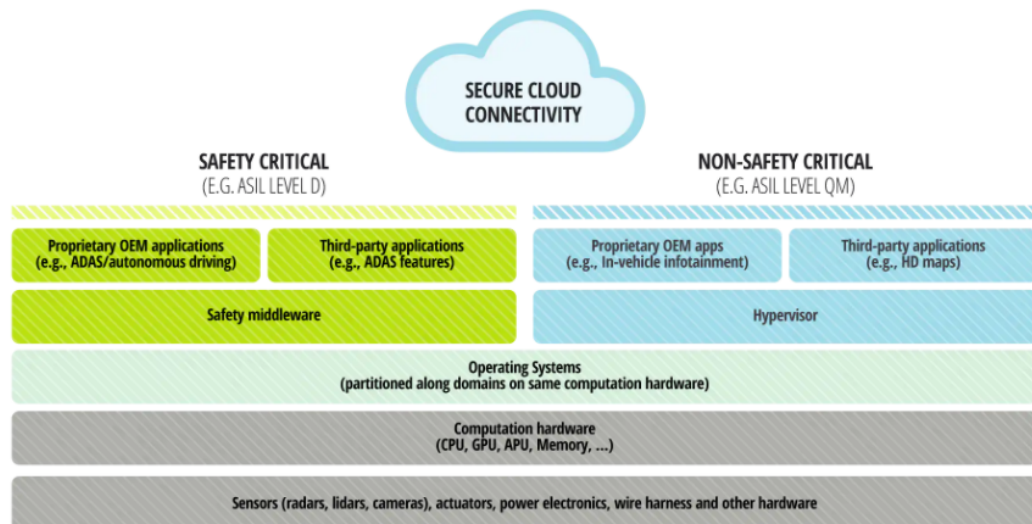
- Time-driven stack. *In this domain, the controller is directly connected to a sensor or actuator while the systems have to support hard real-time requirements and low latency times ; resource scheduling is time based. This stack includes systems that reach the highest Automotive Safety Integrity Level classes, such as the classical Automotive Open System Architecture (AUTOSAR) domain.*
- Event- and time-driven stack. *This hybrid stack combines high-performance safety applications, for example, by supporting ADAS and HAD capability. Applications and peripherals are separated by the operating system, while applications are scheduled on a time base. Inside an application, scheduling of resources can be based on time or priority. The operating environment ensures that safety-critical applications run on isolated containers with clear separation from other applications within the car. A current example is adaptive AUTOSAR.*
- Event-driven stack. *This stack centers on the infotainment system, which is*

not safety critical. The applications are clearly separated from the peripherals, and resources are scheduled using best-effort or event-based scheduling. The stack contains visible and highly used functions that allow the user to interact with the vehicle, such as Android, Automotive Grade Linux, GENIVI, and QNX.

- Cloud-based (off-board) stack. *The final stack covers and coordinates access to car data and functions from outside the car. The stack is responsible for communication, as well as safety and security checks of applications (authentication), and it establishes a defined car interface, including remote diagnostics.*

<https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture>

<https://www2.deloitte.com/us/en/insights/focus/future-of-mobility/pure-play-software.html>



Source: Deloitte research, Aptiv.

FIGURE 2.2 – Architecture générique

Bla bla bla

2.3 Cyber-security principles

Cyber-security is build on 3 pillars; the CIA triad.

The security goal is to provide safetymeasures to achieve the confidentiality, integrity, and availability (CIA) triad for protection of the overall system along with its peripherals. The triad CIA is as follows :

- *Confidentiality* : The aim of confidentiality is to protect the critical information from unauthorized users. Confidentiality for network security ensures that the critical assets are accessible only to authorize users.
- *Integrity* : This ensures that unauthorized users do not modify or manipulate the data or information during their network transmission.
- *Availability* : The availability is the last component of the CIA triad that represents the real availability of our information. Authentication methods, channel access, and systems all have to function efficiently to prevent the data and make sure that it is available when required. In short, the availability aims to ensure that data and network resources are available when requested by the authorized users.



FIGURE 2.3 – 3 pillars of the security

Besides the CIA triad, Identification, Authentication, Authorization, Auditing and Accounting (called AAAA) also play an important role for controlling the access to the system resources. The AAAA is a term for controlling the access to the system resources, auditing usage, enforcing policies, and offering the details need to charge for services.

- *Identification* : Identification aims of claiming to be an identity when attempting to access a resource. Providing an identity can involve typing or sending a username or an ID, swiping a smart card, waving a proximity device Without an identity the system has no way to correlate an authentication factor to the subject.
- *Authentication* : Authentication is about proving that you are that claimed identity. It requires the subject to provide additional information that correspond to the identity that is claimed. The most common form is to provide a password. Authentication verifies the identity of the subject by comparing one or more factors against the database of valid identities.
- *Authorization* : Authorization is defining the permissions of a resource/object access for a specific identity. It ensures that the access to a resource/object is given the right and privileges assigned to the authenticated identity. If the requested action is allowed, the subject is authorized, instead the subject is denied. It is not because a subject is correctly authenticated that he/she has the right to perform any actions on any resource.
- *Auditing* : Auditing is recording a log of events and activities related to the system, subjects and objects. It purposes to track and record all subject requests and actions. Log files provide an audit trail for re-creating the history of events. It permits to detect malicious actions, system failures but also system performances
- *Accounting* : Accountability aims to reviewing the log files to check for compliance or violation of security policy in order to hold the subject accountable of his/her/its actions. It is also a way of evaluating what services have been used and how many resources have been consumed.

Les principales techniques utilisées pour garantir ces 3 propriétés sont :

- *Chiffrement (Encryption)* : La transformation de l'information originale (appelé clair) à l'aide d'une clé de chiffrement, de telle sorte que les informations transformées (appelé chiffré) ne puissent être interprétées que par un autre utilisateur ayant connaissance de la clé de déchiffrement (qui peut, dans certains cas, être identique à la clé de chiffrement). Pour être en sécurité, un algorithme de chiffrement doit rendre extrêmement difficile pour quelqu'un de déterminer tout ou partie des informations claires sans connaissance de la clé de déchiffrement ou faiblesse de l'algorithme de chiffrement.
- *Authentification (Authentication)* : La détermination de l'identité qu'un sujet (personne, logiciel ou équipement). Cette détermination peut être effectuée par plusieurs moyens. Il est généralement basé sur une combinaison de quelque chose que la personne connaît (Something you know) comme un mot de passe ou un code PIN, quelque chose le sujet possède (Something you have) comme une carte à puce contenant des clés secrètes un téléphone pour recevoir un code, un passeport, ou quelque chose la personne est (Something you are) comme une empreinte digitale.
- *Contrôles d'accès (Access Control)* : Ensemble de règles et politiques de sécurité qui limitent l'accès aux informations aux sujets (personnes et/ou systèmes) ayant un *besoin d'en connaître*. Ce besoin d'en connaître est déterminé suite à l'authentification correcte du sujet, de par son identité ou rôle. Un ensemble de règles est pré-défini par le gestionnaire de la sécurité informatique en se basant sur la politique de sécurité.
- *Signature (Signature)* : La signature d'une information a deux objectifs ; assurer que l'information signée n'a pas été altérée depuis la signature et authentifier la source de l'information (non-repudiation). La signature consiste en un chiffre dépendant d'une clé secrète connue uniquement par le sujet signant l'information et du contenu du message à signer. Une signature est vérifiable, sans connaître la clé secrète, par une partie tierce en cas de litige entre parties. Si le contenu du message ou la signature est modifié, la correspondance entre le contenu initial et message et sa signature sera invalide permettant de détecter l'altération et de rejeter l'information. De façon, symétrique, si le contenu du message et sa signature correspondent alors la source de l'information ne pourra pas nier avoir signé l'information car elle est la seule à connaître la clé secrète.
- *Responsabilité (Accountability)* : La capacité de rendre le sujet responsable de ces actions. Ceci est réalisé en s'appuyant sur des journaux d'audit (audit log). Une fois le sujet correctement authentifié, toutes ces actions sont enregistrées sous forme d'événements dans un journal d'audit. En cas d'investigation ou de façon périodique, le gestionnaire de la sécurité informatique peut réaliser un audit des journaux pour identifier la source potentielle d'une attaque.
- *Sensibilisation à la sécurité (Security awareness)* : La principale source de risque pour une organisation ne provient pas de faiblesse dans la technologie des équipements mais d'actions (ou d'inaction) de la part des utilisateurs du système. Afin de limiter ce risque, il est nécessaire de former les utilisateurs.

aux différents risques informatiques et aux bonnes pratiques pour assurer le niveau de sécurité requis.

- *Sécurité physique (physical security)* : Mise en place de barrières physiques pour limiter l'accès aux ressources sensibles. Ces barrières comprennent sont multiples comme le gardiennage, la vidéo-surveillance, les serrures sur les armoires et les portes, les chambres fortes, l'utilisation de matériaux insonorisants, ou même la construction d'équipement renforcé (tempest) afin que les signaux électromagnétiques ne puissent pas entrer ou sortir.
- *Tolérance aux fautes (fault tolerance)* : Ensemble de techniques peuvent être utilisées pour garantir le système en opération.

	Confidentialité	Intégrité	Disponibilité
Chiffrement	✓		
Authentification	✓	✓	
Contrôles d'accès	✓	✓	
Signature		✓	
Responsabilité	✓	✓	
Sensibilisation à la sécurité physique	✓	✓	✓
Sécurité physique	✓	✓	✓
Tolérance aux fautes			✓

TABLE 2.2 – Principales méthodes de la sécurité informatique

Dans le reste de la thèse, nous concentrerons sur la disponibilité des systèmes embarqués.

2.4 Le risque cybersécurité pour l'automobile

L'ajout de logiciels et de connectivité....

3 niveaux d'attaques :

- *Attaques physiques.*
 - Attaque via la prise diagnostique de la voiture. L'objectif est de pouvoir modifier les caractéristiques de la voiture et/ou de rajouter des options sur la voiture. Les logiciels d'une voiture sont hautement configurables. Un même logiciel est installé sur plusieurs gammes de véhicules d'un constructeur. La différenciation entre les deux modèles s'effectue par le paramétrage du logiciel. Si un attaquant a la possibilité de modifier ces paramètres, il peut activer des fonctions optionnelles de la voiture ou modifier les caractéristiques moteur pour booster le véhicule.
 - Attaque via la prise USB de la voiture. L'objectif est de pouvoir créer un point d'entrée pour une attaque courte ou longue portée sur le véhicule. La prise USB peut être connectée sur un équipement radio qui permet d'étendre la portée de l'attaque.
- *Attaque courte portée* L'objectif est de pouvoir prendre le contrôle du véhicule, d'envoyer de fausses informations ou de bloquer les communications aux

véhicules aux alentours. De plus en plus de voitures ont un système d'ouverture de portes et de démarrage sans clé. Par exemple, votre voiture est garée devant votre maison et les clés sont sur le petit d'entrée. Un attaquant peut insérer un équipement radio entre la clé et la voiture (attaque par relais). Un côté est proche de la voiture, l'autre côté est connecté à une antenne scannant les fréquences radio de la clé. Ceci permet d'ouvrir et/ou de démarrer la voiture même quand les clés du véhicule sont hors de portée de la voiture. Une fois que l'antenne a accroché la fréquence de la clé, elle relaye son signal sur l'équipement proche du véhicule. La voiture a ainsi l'impression que le propriétaire est proche et ouvre les portes et autorise le démarrage. Le véhicule peut être volé sans effraction (voir https://www.youtube.com/watch?v=_cua7BFX-Qk pour une vidéo d'attaque relais). Un autre exemple, consiste à brouiller le signal pour la fermeture centralisée d'un véhicule. Le conducteur appuie sur le bouton de fermeture centralisée de sa clé, il a le sentiment que sa voiture est correctement fermée. Mais si le signal est brouillé, la voiture est ouverte et un attaquant peut facilement ouvrir une porte pour voler des affaires dans la voiture.

- *Attaque longue portée.* L'objectif est de prendre le contrôle à distance d'un véhicule. Un des fameux exemples de ce type d'attaque est la prise de contrôle d'une Jeep Cherokee par 2 attaquants ; Andy Greenberg conduisait sa voiture sur l'autoroute vers Saint-Louis, roulant 70 mph. 2 hackers Charlie Miller et Chris Valasek sont installés dans leur canapé avec leur laptop ouvert. Dans un premier temps, la climatisation de la voiture s'est affolée, puis, une image des 2 hackers est apparue sur l'écran de la voiture, la radio a changé de station et le volume a fortement augmenté, Mr Greenberg ne pouvait pas contrôler le volume de la radio ni la station. Ensuite la voiture s'est arrêtée toute seule. (voir MILLER et VALASEK 2015 pour le détail de l'attaque)

Chapitre 3

La théorie des jeux

Il est très important, pour celui qui souhaite découvrir, de ne pas limiter son esprit à un seul chapitre de la science mais plutôt de rester en contact avec plusieurs autres.

Jacques Hadamard.

3.1 Présentation

La théorie des jeux est la science de la prise de décision stratégique. La théorie des jeux a été utilisée dans des sciences aussi diverses que la biologie évolutionniste, le management des décisions (politique) et l'économie. Elle peut être définie comme l'étude de modèles mathématiques des conflits et coopérations entre décideurs intelligents et rationnels. La théorie des jeux fournit des techniques mathématiques générales pour analyser des situations dans lesquelles deux personnes ou plus prennent des décisions qui s'influencent mutuellement.

Dans le langage de la théorie des jeux, un jeu fait référence à toute situation impliquant deux sujets ou plus. Les sujets impliqués dans un jeu peuvent être appelés les *joueurs*. Comme indiqué dans la définition ci-dessus, il y a deux hypothèses de base que les théoriciens des jeux font généralement à propos des joueurs : ils sont rationnels et intelligents. Chacun de ces adjectifs est utilisé ici dans un sens technique qui nécessite quelques explications. Un décideur est rationnel s'il prend des décisions de manière cohérente de ses propres objectifs. En théorie des jeux, en s'appuyant sur le fondamental résultats de la théorie de la décision, nous supposons que l'objectif de chaque joueur est de maximiser la valeur attendue de son propre gain, qui est mesuré en une certaine échelle d'utilité. L'idée derrière un *décideur rationnel* est que les actions sélectionnées (nommées *stratégies*) maximiseront les bénéfices d'utilité attendus par le joueur. Cette idée remonte au moins à Bernoulli (1738), mais la justification moderne de cette idée tient à Von Neumann et Morgenstern (1947) ([xx]).

3.2 Définition générale

Pour expliquer les concepts de la théorie des jeux, prenons un exemple ; *"l'évitement de la congestion du réseau internet"*. Le réseau internet est basé sur le protocole TCP/IP. En TCP/IP, un fichier est découpé en paquets qui transitent entre les différents noeuds du réseau entre l'émetteur et le récepteur. Chaque fois que le récepteur reçoit un paquet, il envoie un accusé de réception à l'émetteur. De cette façon, le l'émetteur sait que le paquet est bien arrivé. Le protocole TCP/IP cherche à augmenter le débit du réseau jusqu'à sa saturation. Lorsque l'un des noeuds du réseau est saturé, il efface des paquets jusqu'à désaturer. L'émetteur ne recevant plus d'accusé réception pour un paquet va le ré-émettre après une certaine temporisation (géré par l'algorithme d'évitement de congestion). Si tous les émetteurs suivent cette règle, cela permet de désaturer le noeud pour le bénéfice de l'ensemble des utilisateurs. Cependant, il est possible pour certains émetteurs de violer cette règle et de ré-émettre le message sans latence.

Ce comportement peut être modélisé en théorie des jeux. Imaginons, deux personnes utilisant Internet. Elles ont deux choix possibles :

- Utiliser la version correct de l'algorithme d'évitement de congestion du réseau. Cette stratégie est notée C .
- Utiliser une version défectueuse de l'algorithme d'évitement de congestion du réseau. Cette stratégie est notée D .

Le comportement du réseau est le suivant ; si les 2 personnes choisissent la stratégie C , chaque paquet sera retardé de 2ms. Si les 2 personnes choisissent la stratégie D , chaque paquet sera retardé de 4ms. Si une des personnes choisit l'action C et l'autre l'action D , le premier voit ses paquets retardés de 6ms et le second voit ses paquets reçus sans retard. Bien sûr, chaque personne est rationnel et cherche à augmenter son débit, c'est à dire dans notre exemple, à minimiser le retard de ses paquets.

3.2.1 Types de jeux

Jeux simultanés

Jeux tour par tour

Jeux bayésien

Jeux stochastique

Jeux dynamiques

Jeux à apprentissage

Définition 3.1. *Un jeu peut être modélisé par un triplet $\langle N, S, u \rangle$ avec :*

- N représente les joueurs. $N = \{1, 2, \dots, n\}$ est un ensemble de cardinal n (nombre de joueurs dans le jeu).
- S représente l'ensemble des combinaisons de stratégies possibles pour l'ensemble des joueurs $S : S_1 \times S_2 \times \dots \times S_n$ avec S_i l'ensemble des stratégies possibles pour le joueur i . Tous les joueurs n'ont pas nécessairement les mêmes stratégies et certaines combinaisons de stratégies peuvent être impossibles.

- u représente la fonction de $u : S \rightarrow \mathbb{R}^n$ qui associe à chaque combinaison de stratégie une valeur d'utilité avec pour chaque u_i :
 - une valeur négative représentant une perte pour le joueur i
 - une valeur positive représentant un gain pour le joueur i

Il est plus commode de représenter la fonction d'utilité pour le joueur i comme $u(s_i, s_{-i})$, avec $s_{-i} = (s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ qui représente la stratégie jouée par tous les autres joueurs et de noter u_i la i -ème composante de la fonction d'utilité.

Pour notre exemple, nous avons :

- 2 joueurs, $N = \{1, 2\}$
- les stratégies possibles pour les 2 joueurs sont identiques $S_1 = S_2 = \{C, D\}$. Et l'ensemble des combinaisons de stratégies possibles est défini comme $S = \{(C, C), (D, C), (C, D), (D, D)\}$
- la fonction utilité peut être représentée par la matrice suivante :

$Joueur_1 \backslash Joueur_2$		C	D
		C	D
C		$(-2, -2)$	$(-6, 0)$
D		$(0, -6)$	$(-4, -4)$

3.3 Stratégie pure

3.3.1 Dominance

Intro Dominance

Definition 3.2. Une stratégie s_i^* domine strictement (*strictly dominates*) la stratégie s_i ssi :

$$\forall s_{-i} \in S_{-i}, u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i})$$

Quelque soit ce la stratégie choisie que les autres joueurs, la stratégie s_i^* est le meilleur choix pour le joueur i . Comme le joueur i est rationnel, il ne va jamais jouer une stratégie dominée s_i .

Definition 3.3. Une stratégie s_i^* domine faiblement (*weakly dominates*) la stratégie s_i , ssi

$$\forall s_{-i} \in S_{-i}, u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$$

et

$$\exists s_{-i} \in S_{-i}, u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i})$$

Quelque soit la stratégie choisie par les autres joueurs, la stratégie s_i^* est au moins aussi bonne que la stratégie s_i pour le joueur i .

Il est maintenant possible d'étendre ces définitions sur l'ensemble des stratégies possibles pour le joueur i .

Definition 3.4. Une stratégie s_i^d est strictement dominante (*resp.* faiblement dominante) pour le joueur i ssi s_i^d domine strictement (*resp.* domine faiblement) toutes les autres stratégie pour le joueur i .

Pour notre exemple, pour le joueur 1, la stratégie D domine strictement la stratégie C . En effet, on a $S_{-1} = \{C, D\}$. pour $s_{-1} = C, u_1(D, C) > u_1(C, C)$ et pour $s_{-1} = D, u_1(D, D) > u_1(C, D)$. En fait, la stratégie D est également la stratégie strictement dominante pour le joueur 2.

Comme le joueur i est rationnel, et qu'il existe une stratégie strictement dominante, alors le joueur i ne va pas jouer une autre stratégie.

3.3.2 Équilibre

Definition 3.5. *Un profil de stratégie (strategy profile) $S^d = (s_1^d, s_2^d, \dots, s_n^d)$ est un équilibre strictement dominant ssi s_i^d est une stratégie dominante pour chaque joueur i .*

Pour notre exemple, $S^d = (D, D)$ est un équilibre strictement dominant. Il est intéressant de remarquer l'importance des valeurs de la fonction d'utilité. Si on prend l'hypothèse que le réseau s'écroule dans le cas où les 2 joueurs suivent la stratégie défectueuse (ie $u(D, D) = (-8, -8)$). Alors il n'y a plus de stratégie dominante pour chaque joueur, et donc plus d'équilibre strictement dominant.

Definition 3.6. *Pour le joueur i , une stratégie s_i est nommée meilleure réponse (Best Response) pour le profil de stratégie σ_{-i} , ssi*

$$\forall s'_i \in S_i, u_i(s_i, \sigma_{-i}) \geq u_i(s'_i, \sigma_{-i})$$

On note $MR(\sigma_{-i})$, l'ensemble des meilleures réponses en face de σ_{-i} .

Definition 3.7. *Un profil de stratégie $s^* = (s_1^*, s_2^*, \dots, s_n^*)$ est une équilibre de Nash ssi pour chaque joueur i , s_i^* est la meilleure réponse.*

3.4 Stratégie mixte

Chapitre 4

Expression de la résilience

Il est très important, pour celui qui souhaite découvrir, de ne pas limiter son esprit à un seul chapitre de la science mais plutôt de rester en contact avec plusieurs autres.

Jacques Hadamard.

4.1 Problématique

4.2 Changement d’adressage dynamique

Le pourquoi, Lla solution proposée, les problèmes identifiés Le papier 1.

4.3 La théorie des jeux

Sélection du type de jeux, le modèle sélectionné, les problèmes identifiés

4.4 Apprentissage

Bla bla bla Attaquant non rationel Papier 2

A	B	C
1	2	3
4	5	6

TABLE 4.1 – very basic table

Chapitre 5

Conclusions et perspectives

*Je vois refléter dans mon miroir tout mon
passé et tout mon avenir.*

J. Cortázar.

5.1 Les leçons apprises

Aucune !

5.2 Les frontières de la recherche

The sky is the limit...

Acronymes

API Application Programming Interface. 10

Glossaire

Threat Potential cause of compromise of cybersecurity properties of one or more assets in order to realize a damage. 10

Bibliographie

- AUTOSAR (2003). *Platform solutions*. URL : <http://www.autosar.org/standards>.
- CHARETTE, Robert N. (2009). “This car runs on code”. In : *IEEE Spectrum* 7649.
- MILLER, Charlie et Chris VALASEK (2015). *Remote exploitation of an unaltered passenger vehicle*. URL : <http://www-cs-faculty.stanford.edu/~uno/abcde.html>.
- STANDARDIZATION, International Organization for (2018). “26262”. In : *Road Vehicles Functional Safety Package*.
- AUTOSAR (2020). *Layered Software Architecture*. URL : https://www.autosar.org/fileadmin/user_upload/standards/classic/20-11/AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf.
- STANDARDIZATION, International Organization for (2021). “21434”. In : *Road Vehicles Cybersecurity engineering*.

Annexe A

Ma belle annexe

A.1 Introduction

La théorie....

Ceci est une appendice...