الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدةد. مولاي الطاهر

كلية الرياضيات و الإعلام الآلي و الاتصالات السلكية و اللاسلكية

قسم: الإعلام الآلي

Mémoire de Master en informatique

Spécialité : Réseau Informatique et Système Répartie

Thème

# Amélioration de la sécurité dans l'IoT à l'aide de l'algorithme AES modifié

- **Présenté par :**

Khelef Abdelkrim

Soltani Mohamed Habib

- **Dirigé par :**

Dr Taleb Fadia

Année universitaire 2024-2025

## ملخص

في أنظمة إنترنت الأشياء، تُصعّب الموارد المحدودة استخدام أساليب تشفير معقدة. ولمواجهة هذا التحدي، يقترح هذا العمل نسخةً خفيفة الوزن منAES ، تُسمىAES-Chaos ، تدمج حزمة الخدمات اللوجستية لتوليد مفاتيح فرعية ديناميكيًا. يُعزز هذا النهج الأمان (NPCR و UACI) مع تقليل وقت التنفيذ باستخدام 8 جولات فقط.

وبالتالي، يُقدّم حلاً فعّالاً مُلائمًا لقيود الكائنات المتصلة.


الكلمات المفتاحية:

Internet des Objets (IoT), Sécurité,AES, AES modifié, Cryptographie légère, suite logistique, Systèmes chaotiques,. Consommation énergétique. Débit, Latence, Clé dynamique, Vecteur d'initialisation (IV), Algorithme de chiffrement,NPCR,UACI



## Abstract

In IoT systems, limited resources make it difficult to use heavyweight encryption methods. To address this challenge, this work proposes a lightweight version of AES, called AES-Chaos, integrating the logistics suite to dynamically generate subkeys. This approach strengthens security (NPCR, UACI) while reducing execution time by using only 8 rounds. It thus offers an efficient solution adapted to the constraints of connected objects.


Keywords:

Internet des Objets (IoT), Sécurité,AES, AES modifié, Cryptographie légère, suite logistique, Systèmes chaotiques,. Consommation énergétique. Débit, Latence, Clé dynamique, Vecteur d'initialisation (IV), Algorithme de chiffrement,NPCR,UACI



## Résumé

Dans les systèmes IoT, les ressources limitées rendent difficile l'usage de méthodes de chiffrement lourdes. Pour répondre à ce défi, ce travail propose une version légère de l'AES, appelée AES-Chaos, intégrant la suite logistique pour générer dynamiquement les sous-clés. Cette approche renforce la sécurité (NPCR, UACI) tout en réduisant le temps d'exécution grâce à l'utilisation de 8 tours seulement. Elle offre ainsi une solution efficace et adaptée aux contraintes des objets connectés.


Mots clés:

Internet des Objets (IoT), Sécurité,AES, AES modifié, Cryptographie légère, suite logistique, Systèmes chaotiques,. Consommation énergétique. Débit, Latence, Clé dynamique, Vecteur d'initialisation (IV), Algorithme de chiffrement,NPCR,UACI

# REMERCIEMENT

En préambule à ce mémoire nous remercions ALLAH qui nous a aidé et nous a donné la patience et le courage durant ces langues

années d'étude.

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude. Je voudrais tout d'abord adresser toute ma reconnaissance à la directeur de ce mémoire, Dr. Taleb Fadia, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui

ont contribué à alimenter ma réflexion.

Je désire aussi remercier les professeurs de l'université de Saida de la département informatique, qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires. Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout

au long de ma démarche.

Nous tenons encore à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences

nous ont soutenu dans la poursuite de nos études.

Enfin, nous remercions toute personne qui a participé de près ou de loin pour l'accomplissement de ce modeste travail.

# Table des matières

# Table des figures

# Liste des tableaux

# Liste des abréviations

**IoT** Internet des Objets

**AES** Advanced Encryption Standard

**CBC** Cipher Block Chaining

**ECB** Electronic Codebook

**CTR** Counter Mode

**Sbox** Substitution Box

**SPN** Substitution-Permutation Network

**CPU** Central Processing Unit

**RAM** Random Access Memory

**IV** Initialization Vector

**TLS** Transport Layer Security

**SSL** Secure Sockets Layer

**UACI** Unified Average Changing Intensity

**NPCR** Number of Pixels Change Rate

**VPN** Virtual Private Network

# Introduction Générale

L'avènement de l'Internet des Objets (IoT) a profondément transformé notre manière d'interagir avec l'environnement numérique. En reliant des dispositifs physiques à Internet, l'IoT permet de collecter, analyser et transmettre des données en temps réel, ouvrant ainsi la voie à des applications intelligentes dans divers domaines tels que la santé, la domotique, les villes intelligentes, l'industrie 4.0 ou encore les transports.

Cependant, cette interconnectivité massive soulève de nouveaux défis en matière de sécurité et de protection des données . Les dispositifs IoT sont souvent déployés dans des environnements ouverts, vulnérables à divers types d'attaques. De plus, ces appareils disposent généralement de ressources matérielles limitées (processeur, mémoire, énergie), ce qui rend difficile l'intégration de mécanismes de sécurité classiques trop coûteux en termes de calcul ou de consommation énergétique.

Dans ce contexte, la cryptographie joue un rôle fondamental pour assurer la confidentialité , l'intégrité et l'authenticité des données transmises entre les objets connectés. L'algorithme AES (Advanced Encryption Standard) est aujourd'hui l'un des standards les plus utilisés grâce à sa robustesse et son efficacité. Cependant, son application directe dans les environnements IoT peut s'avérer inadaptée en raison de ses exigences en ressources.

Pour pallier ces limitations, une solution prometteuse consiste à renforcer AES par des techniques issues de la théorie du chaos . Ces systèmes dynamiques non linéaires présentent des propriétés intéressantes telles que la sensibilité aux conditions initiales , l'imprévisibilité , ce qui les rend particulièrement adaptés aux architectures embarquées.

Ainsi, cette recherche explore une approche innovante visant à modifier l'algorithme AES standard en y intégrant des éléments chaotiques, notamment via la fonction logistique , afin de générer de manière dynamique des clés de chiffrement et des vecteurs d'initialisation. Cette méthode vise à garantir un bon compromis entre sécurité accrue , faible coût computationnel et efficacité énergétique , tout en restant compatible avec les contraintes inhérentes aux dispositifs IoT.

# Chapitre 1

# Sécurité de l'Internet des Objets

## 1.1 Introduction

L'Internet des objets (IoT) désigne un réseau d'objets physiques intégrés à des capteurs, des logiciels et d'autres technologies permettant leur interconnexion via Internet. Cette évolution technologique a révolutionné divers secteurs tels que la santé, l'industrie, l'agriculture et les maisons intelligentes, apportant de nombreux avantages en termes de confort, de gestion et d'efficacité. Cependant, cette interconnexion massive soulève des préoccupations majeures en matière de sécurité. En effet, la multiplication des objets connectés, souvent dépourvus de mécanismes de sécurité solides, augmente les risques d'attaques et de violations de données. Des études ont montré que la majorité des objets IoT souffrent de vulnérabilités liées à des failles de conception, de communication ou de gestion des mots de passe (Sicari, Coen-Porisini, De Pellegrini  Miorandi, 2015) **1**. De plus, les objets connectés, souvent peu sécurisés, peuvent servir de points d'entrée pour des attaques ciblant des systèmes plus sensibles, comme des infrastructures critiques ou des réseaux privés (Zhou, Cao, Dong  Vasilakos, 2018) **2**.

Les risques associés à l'IoT incluent des attaques par déni de service distribué (DDoS), des vols de données personnelles, des intrusions non autorisées et la manipulation malveillante des dispositifs. Un exemple frappant est l'attaque Mirai en 2016, où des dispositifs IoT infectés ont été utilisés pour mener une attaque DDoS de grande envergure (Kaspersky, 2017) **3**. En outre, la gestion de la sécurité des objets connectés est compliquée par la diversité des protocoles utilisés, la rareté des mises à jour de sécurité et l'absence de normes globales sur la cyber sécurité de l'IoT (Weber, 2010) **4**. La sécurité des objets connectés implique donc la mise en place de solutions de cryptographie, d'authentification forte, ainsi que de protocoles de communication sécurisés, afin de garantir la confidentialité et l'intégrité des données échangées (Roman, Zhou,  Lopez, 2013 ) **5**. Les chercheurs soulignent également la nécessité d'une gouvernance adéquate pour superviser l'ensemble des dispositifs IoT et la mise en œuvre de politiques de sécurité rigoureuses dès la phase de conception des objets (Gubbi, Buyya, Marusic, Palaniswami, 2013 ) **6**.

Il devient donc impératif de renforcer la sécurité de l'IoT à travers des approches novatrices et de sensibiliser les utilisateurs et les développeurs aux enjeux associés à la cyber sécurité. L'adoption de bonnes pratiques et de standards de sécurité universels est une étape essentielle pour minimiser les risques et tirer pleinement parti des avantages de cette technologie émergente.

## 1.2 Évolution de l'Internet des Objets

### 1.2.1 Historique de l'IoT

L'Internet des objets (IoT) trouve ses origines dans les années 1990, lorsque les premières expérimentations d'objets connectés ont émergé, tels que des grille-pain et des machines à café intégrant une connexion numérique pour automatiser certaines fonctionnalités (Ashton, 2009) **1**. En 2000, le fabricant coréen LG a été l'un des premiers industriels à envisager sérieusement le développement d'électroménagers connectés à Internet, ouvrant la voie aux premières expérimentations d'appareils capables de rechercher automatiquement des informations en ligne (Gubbi et al., 2013) **2**.

En 2003, alors que la population mondiale atteignait environ 6,3 milliards, on comptait environ 500 millions d'appareils connectés à Internet (Cisco IBSG, 2011) **3**.

Le ratio appareils/population (0,08) indiquait alors un faible taux de connexion par habitant. Selon la définition de Cisco IBSG, l'Internet des objets tel qu'il est défini aujourd'hui n'existait pas encore, car la proportion d'objets connectés était insuffisante pour représenter un véritable écosystème numérique (Evans, 2011) **4**.

Grâce à l'essor des smartphones et des tablettes, le nombre d'appareils connectés a fortement augmenté, atteignant 12,5 milliards en 2010, alors que la population mondiale était de 6,8 milliards d'habitants (Evans, 2011) **4**. Pour la première fois, le nombre d'objets connectés par personne dépassait 1,84.

Aujourd'hui, ce chiffre continue de croître exponentiellement, dépassant largement la population mondiale. Les estimations prévoient qu'il pourrait atteindre 50 milliards d'objets connectés d'ici 2030, redéfinissant ainsi les usages numériques et les infrastructures technologiques (Cisco IBSG, 2011) **3**.

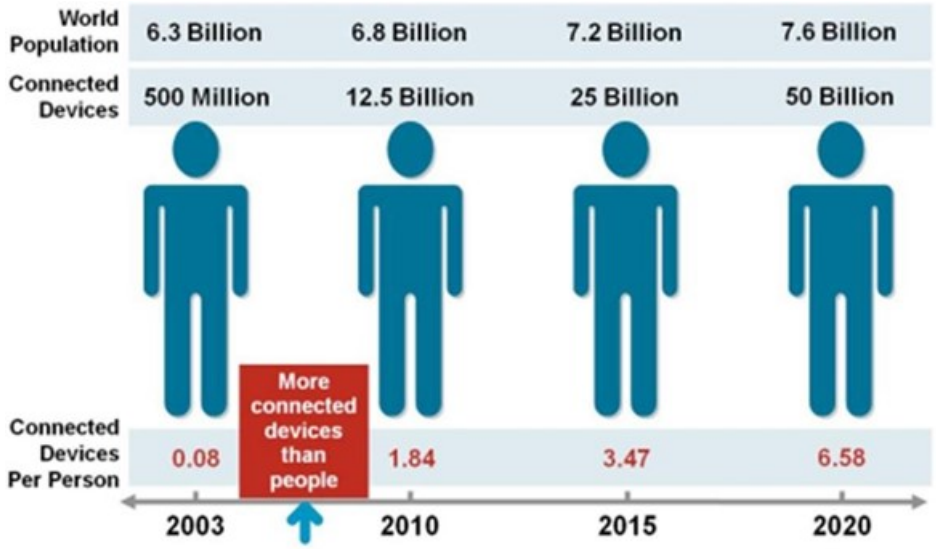



FIGURE 1.1 – IoT Aujourd'hui


FIGURE 1.2 – Future de l'IoT

### 1.2.2 De l'Internet à l'Internet des Objets

L'Internet a émergé dans les années 1960 comme un réseau de communication reliant des ordinateurs pour partager des informations. À ses débuts, il était principalement

utilisé par des chercheurs et des institutions académiques pour l'échange de données et le développement de nouvelles technologies (Leiner et al., 2009) **1**.

Au fil des décennies, Internet s'est progressivement étendu à une échelle mondiale, devenant une infrastructure essentielle pour les communications, le commerce et le divertissement (Kahn  Cerf, 1999) **2**. Cette évolution a posé les bases du développement de l'Internet des objets (IoT), une avancée majeure où des objets physiques – tels que des appareils électroménagers, des véhicules ou des capteurs – sont désormais connectés au réseau pour permettre un échange de données automatisé et en temps réel (Gubbi et al., 2013) **3**.

L'IoT transforme profondément des domaines comme la santé, l'agriculture et l'industrie, en optimisant les opérations et en renforçant l'efficacité des systèmes automatisés (Atzori et al., 2010) **4**. Toutefois, cette interconnexion croissante soulève des enjeux cruciaux en matière de sécurité informatique et de gestion des données, nécessitant de nouvelles solutions cryptographiques et des standards de cybersécurité adaptés aux réseaux distribués (Roman et al., 2011) **5**.

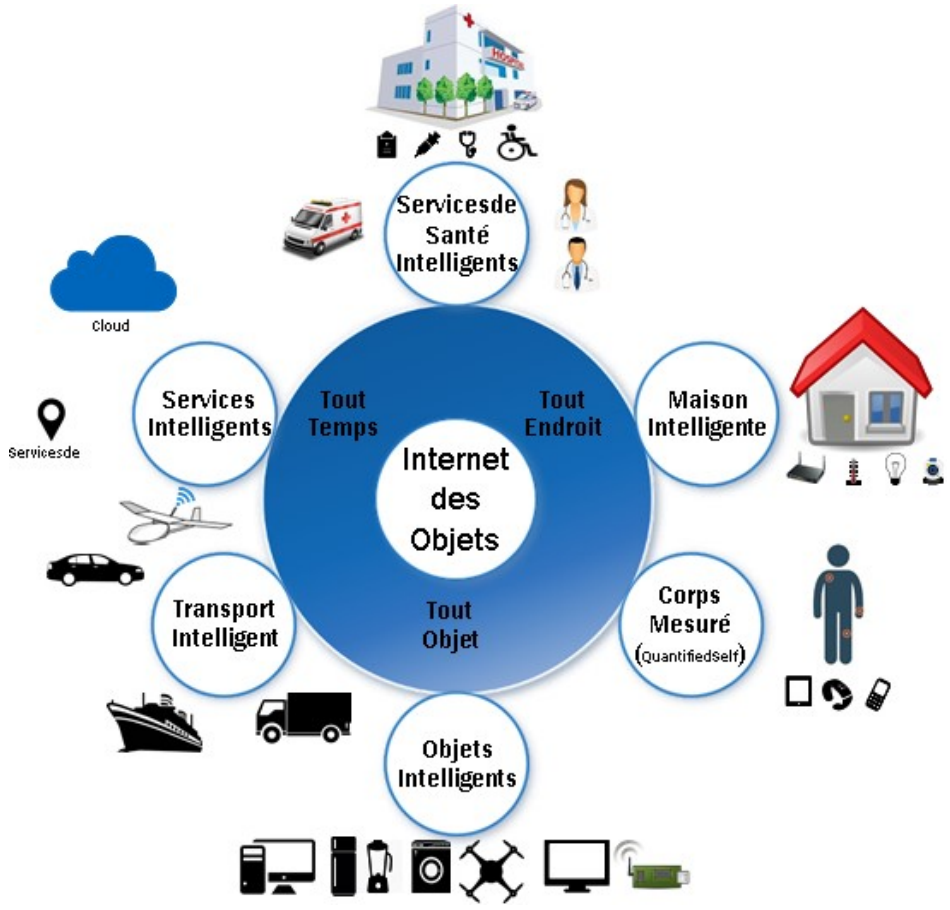



FIGURE 1.3 – Future de l'IoT

## 1.3 Technologies associées à l'IoT

### 1.3.1 Domotique

La domotique regroupe un ensemble de technologies permettant de rendre les maisons intelligentes, capables de fonctionner de manière autonome et de contrôler

divers équipements via une interface centralisée, telle qu'un téléphone ou un panneau de contrôle (Balta-Ozkan et al., 2013) **1**. L'Internet des objets (IoT) joue un rôle clé dans cette évolution, facilitant la communication entre les appareils électroménagers et permettant leur gestion à distance (Reinisch et al., 2011) **2**.

Grâce à ces avancées, la domotique ne se limite plus aux foyers individuels ; elle s'étend désormais aux villes intelligentes, où des infrastructures connectées optimisent la gestion des ressources urbaines, telles que l'éclairage public, les systèmes de chauffage et la sécurité (Chan, Campo, Estève, Fourniols, Escriba  Campo, 2008) **6 3**.

FIGURE 1.4 – Domotique

### 1.3.2 Maisons intelligentes (Smart Homes)

La maison connectée ou intelligente (Smart Home) est une expression pour désigner un habitat où plusieurs éléments sont contrôlés à distance, éventuellement de manière automatisée : le chauffage, la lumière, les alarmes, etc. C'est une avancée de la domotique. Elles intègrent des objets connectés ainsi que leurs technologies aux solutions existantes de la domotique, permettant ainsi une interaction directe avec les utilisateurs (Poussart, 2021) **1**. La principale distinction entre une maison traditionnelle et une maison connectée réside dans le degré de connectivité. Dans une maison traditionnelle dépourvue de technologie intelligente, les appareils fonctionnent de manière locale et manuelle. Pour activer une fonctionnalité spécifique, il est généralement nécessaire d'actionner un interrupteur ou un bouton. Ces appareils dits traditionnels ont des commandes et une gestion de l'énergie limitée. À côté de l'habitation traditionnelle, l'habitation connectée permet une gestion ainsi qu'un contrôle à distance des différents appareils, dits « intelligents ». Cette gestion est facilitée par un réseau de capteurs et d'ordinateurs (Balta-Ozkan, Boteler  Connor, 2013) **2**.

Dans la littérature anglophone, il a été constaté que le terme de "smart home" est un terme qui peut être utilisé pour désigner à la fois une maison connectée et une maison intelligente, selon le niveau de sophistication et d'automatisation du système. Cependant dans la littérature, les termes de maison connectée et de maison intelligente sont bien souvent utilisés de façon interchangeable tandis qu'une différence subtile existe entre les deux. Également, le terme de maison automatique tend à être confondu avec les deux précédents concepts. Il est toutefois important de noter que le terme smart home, est relativement nouveau et en constante évolution.

Ainsi, dans ce mémoire, nous avons tenté de faire une distinction entre ces termes. L'évolution de la maison intelligente peut ainsi être retracée de la façon suivante :

1. **La maison automatique** : Le concept de maison automatisée se réfère à une habitation où tous les équipements et appareils électroménagers sont contrôlés automatiquement. Cette notion a émergé dans les années 80, marquant les premiers pas de la technologie au sein des foyers. Une gamme de capteurs surveille en permanence diverses variables telles que la température, l'humidité et la luminosité. Ces données sont ensuite liées à des scénarios prédéfinis, déclenchant ainsi des actions spécifiques. Par exemple, les éclairages s'allument lorsque l'environnement devient trop sombre, ou le chauffage s'active lorsque la température descend en dessous d'un seuil prédéterminé.
2. **La maison connectée** : c'est une demeure équipée d'appareils reliés à Internet et contrôlables via des applications mobiles ou des assistants vocaux. Les dispositifs connectés peuvent englober des thermostats intelligents, des serrures de porte, des caméras de sécurité, des systèmes de divertissement, etc. L'objectif est de permettre aux résidents de gérer leur domicile aisément et à distance. Cela a engendré la prolifération de solutions de surveillance, permettant de recevoir des alertes sur des smartphones et de visualiser les flux des caméras de vidéosurveillance par exemple.
3. **La maison intelligente** : représente le stade le plus avancé parmi ces avancées technologiques. Elle fusionne les éléments de la domotique, de la maison automatisée et de la maison connectée, mais avec l'incorporation de l'intelligence artificielle. Ainsi, selon les auteurs Kim et al. (2020), la smart home a été étendue en termes d'installation de capteurs dans les objets utilisés quotidiennement et permettant l'interopérabilité avec les appareils mobiles (maison connectée), mais, lorsqu'elle est capable d'apprendre du comportement de ses occupants et des objets intelligents afin de prendre des décisions critiques par elle-même, nous sommes donc dans l'intelligence de la maison.

   Ainsi, les maisons intelligentes utilisent des algorithmes d'apprentissage automatique pour apprendre les habitudes et les préférences des occupants, et ajustent automatiquement les équipements pour répondre à leurs besoins. Les maisons intelligentes peuvent également être interconnectées pour créer des systèmes complexes et automatisés qui anticipent les besoins des occupants. On peut les illustrer par le thermostat Nest par exemple, qui apprend en permanence les habitudes des occupants de la maison pour adapter le chauffage du bâtiment aux rythmes et préférences des différents membres d'une famille.

   En somme, pour qu'une maison soit considérée comme "intelligente", elle doit combiner les avantages de plusieurs appareils intelligents. Un seul appareil en lui-même n'est en fait pas si intelligent ; c'est lorsque cet appareil parle et corrèle des entrées d'autres appareils domestiques, pour calculer et prendre des décisions basées sur les données, que nous atteignons la promesse de domotique et d'intelligence. Cependant, une fois qu'on a fait la liste de ce que la smart home peut faire, on peut raisonnablement se poser la question. Nous sommes aujourd'hui encore très loin d'une véritable intelligence. Aujourd'hui la technologie permet simplement de concevoir une maison connectée. C'est-à-dire une maison que l'on peut piloter à distance depuis son smartphone, mais finalement rien de plus.

   Enfin, il est important de souligner qu'il existe plusieurs définitions au travers de la littérature pour conceptualiser et définir les smart home. Dépendamment de leur champ et de l'objet de leur étude, certains auteurs mettront davantage

d'emphase sur la capacité des smart home à gérer leur consommation d'énergie (Reinisch, Kofler, Kastner  Neugschwandtner, 2011 ; Scott, 2007) **4**, tandis que d'autres mettront en avant leur capacité à répondre aux besoins des résidents à travers une technologie automatisée (Balta-Ozkan, Boteler  Connor, 2013a ; De Silva, Morikawa  Petra, 2012) **5**, voire leur rôle bénéfique en matière de soins de santé pour les utilisateurs vieillissants (Chan, Campo, Estève, Fourniols, Escriba  Campo, 2008) **6**.

Bien que ces définitions sur les smart home diffèrent, elles ont malgré tout trois points communs : la technologie, les services et leur capacité à satisfaire les besoins des utilisateurs (Marikyan, Papagiannidis  Alamanos, 2019) **7**. Pour Marikyan et al. (2019), le mot « smart » est récemment devenu un terme générique pour toute technologie innovante possédant un quelconque degré d'intelligence artificielle. Selon d'autres auteurs, les attributs clés d'une technologie intelligente sont la capacité d'acquérir des informations du milieu environnant et de réagir en conséquence (Chan, Campo, Estève, Fourniols, Escriba  Campo, 2008 ; Balta-Ozkan, Boteler  Connor, 2014) **8**. Ainsi, qu'il s'agisse d'une maison connectée ou d'une maison intelligente, l'objectif à long terme de la technologie intelligente est d'améliorer le bien-être des individus (Hong, Chen, Wang Lou, 2016) **9**, en fournissant des solutions plus pratiques et efficaces pour la gestion de la maison.

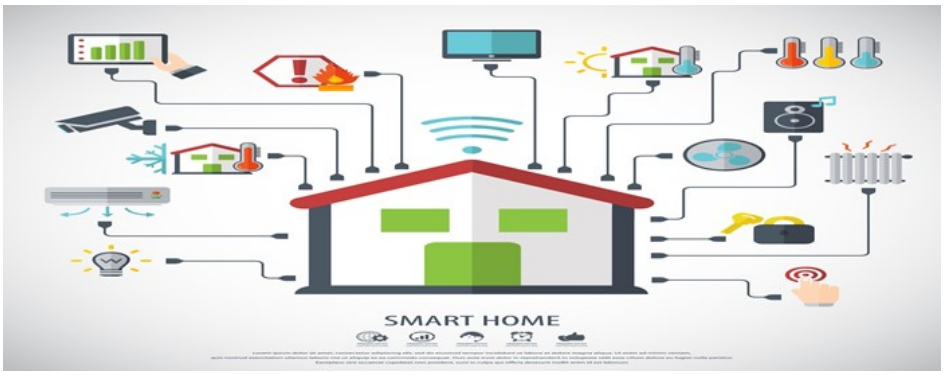



FIGURE 1.5 – SMART HOMES

### 1.3.3 Informatique en nuage (Cloud Computing)

Le Cloud Computing (ci-après Cloud) est défini par le National Institute of Standards and Technology (NIST) comme « un modèle permettant un accès réseau omniprésent, pratique et à la demande à un ensemble partagé de ressources informatiques configurables (par exemple, des réseaux, serveurs, stockage, applications et services) qui peuvent être rapidement approvisionnées et libérées avec un minimum d'efforts de gestion ou d'interaction avec le fournisseur de services » (Mell  Grance, 2011) **1**.

Pour les organisations, la stratégie de migration vers le Cloud consiste à déplacer les applications et environnements informatiques traditionnellement hébergés en interne (on-premises) vers des fournisseurs d'infrastructures cloud (Marinescu, 2013) **2**. Cette approche permet d'externaliser la gestion de tout ou partie des logiciels, applications et services informatiques, offrant ainsi une meilleure scalabilité et une réduction des coûts opérationnels grâce à la mutualisation des ressources (Armbrust et al., 2010) **3**.

Le Cloud offre de nombreuses opportunités, notamment l'accélération de la transformation numérique des organisations publiques et privées, en améliorant la performance

des services, leur disponibilité et la flexibilité des usages (Zhang et al., 2010) **4**. Par ailleurs, la sécurité et la gouvernance des données restent des enjeux cruciaux, nécessitant des politiques strictes de protection des informations sensibles et de gestion des accès (Buyya et al., 2011) **5**.

Ce marché est en pleine expansion, contribuant significativement à la croissance des services numériques. Selon le cabinet Markess by Exaegis, le marché français du Cloud devrait atteindre 27 milliards d'euros d'ici 2025, illustrant l'adoption croissante de ces technologies à l'échelle nationale (Markess, 2023) **6**.

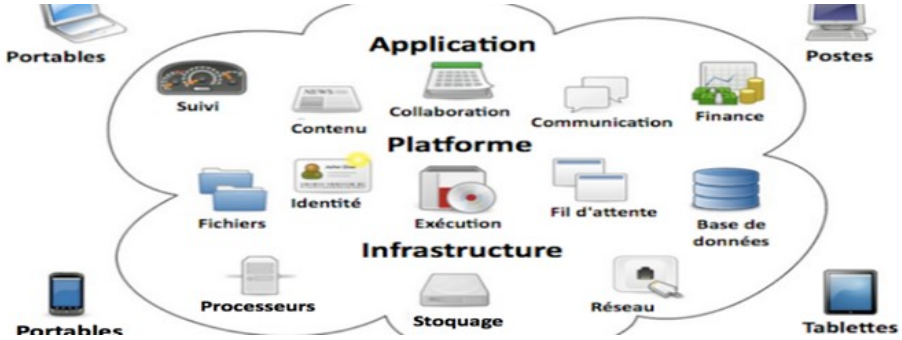



FIGURE 1.6 – Cloud Computing

### 1.3.4 Informatique en proximité (Edge Computing)

L'informatique en proximité, ou Edge Computing, est une architecture distribuée où le traitement des données se fait près de leur source, au lieu d'être systématiquement envoyé vers des centres de données distants. Cette approche permet une réduction significative de la latence, un traitement plus rapide et une meilleure gestion des ressources, ce qui est particulièrement essentiel pour des applications nécessitant des réponses en temps réel (Shi et al., 2016) **1**.

L'objectif principal du Edge Computing est de rapprocher le calcul et le stockage des utilisateurs finaux ou des dispositifs IoT (Internet des Objets), garantissant ainsi une exécution efficace des tâches critiques et sensibles au temps (Satyanarayanan, 2017) **2**. Cette technologie est particulièrement utile dans des environnements industriels, pour les véhicules autonomes, ainsi que pour des applications à forte exigence de réactivité, telles que les réseaux de télécommunications 5G et les systèmes de santé connectés (Gai et al., 2020) **3**.

En optimisant les réseaux et les ressources locales, l'Edge Computing permet de traiter une grande quantité de données sans surcharger le réseau principal, améliorant ainsi la scalabilité et l'efficience énergétique des systèmes informatiques modernes (Xu et al., 2018) **4**.

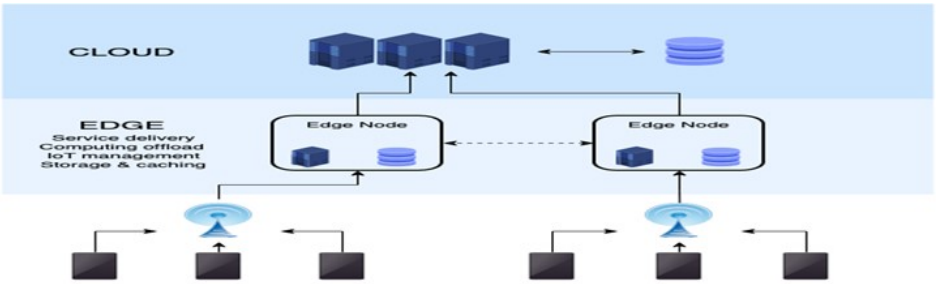



FIGURE 1.7 – Edge computing

## 1.4 Défis de l'Internet des Objets

L'Internet des Objets (IoT) fait face à plusieurs défis majeurs. D'abord, la sécurité est une priorité, car les objets connectés sont vulnérables aux cyberattaques, ce qui peut compromettre la vie privée et l'intégrité des données. Ensuite, il y a la gestion des données, car l'IoT génère d'énormes volumes d'informations, nécessitant des systèmes efficaces pour leur stockage et traitement. La compatibilité entre les appareils pose également un problème, car il existe une multitude de normes et de protocoles. La consommation énergétique des dispositifs IoT est un autre défi, surtout pour les appareils mobiles ou distants (Shi et al., 2016) **4**. La connectivité peut être instable ou insuffisante, rendant l'IoT moins fiable dans certaines régions. En outre, le coût de déploiement de ces technologies reste élevé. L'évolutivité des solutions IoT face à l'augmentation continue d'appareils est également un enjeu. Le respect de la réglementation en matière de données personnelles est crucial pour assurer la conformité. Enfin, l'interopérabilité des systèmes IoT est essentielle pour assurer une communication fluide entre différents appareils.

### 1.4.1 Hétérogénéité des dispositifs

L'hétérogénéité des dispositifs désigne la diversité des technologies, des protocoles et des plateformes utilisées dans les systèmes connectés. Cette variabilité rend l'intégration et l'interopérabilité des dispositifs IoT particulièrement complexes. Les appareils peuvent être très différents en termes de capacités, de performances, de systèmes d'exploitation et de normes de communication (Xu et al., 2018) **9**, ce qui crée des défis pour assurer leur communication et leur gestion au sein d'un même réseau. L'hétérogénéité affecte la compatibilité des logiciels et matériels, nécessitant des solutions flexibles pour assurer leur interaction. De plus, elle complique le développement de solutions universelles ou standardisées. Le défi réside aussi dans la gestion des mises à jour et la maintenance de ces appareils, parfois très variés. Pour garantir la sécurité, chaque type de dispositif nécessite une approche adaptée. Enfin, cette diversité nécessite un contrôle constant de la performance et de la fiabilité des systèmes pour répondre aux besoins des utilisateurs.

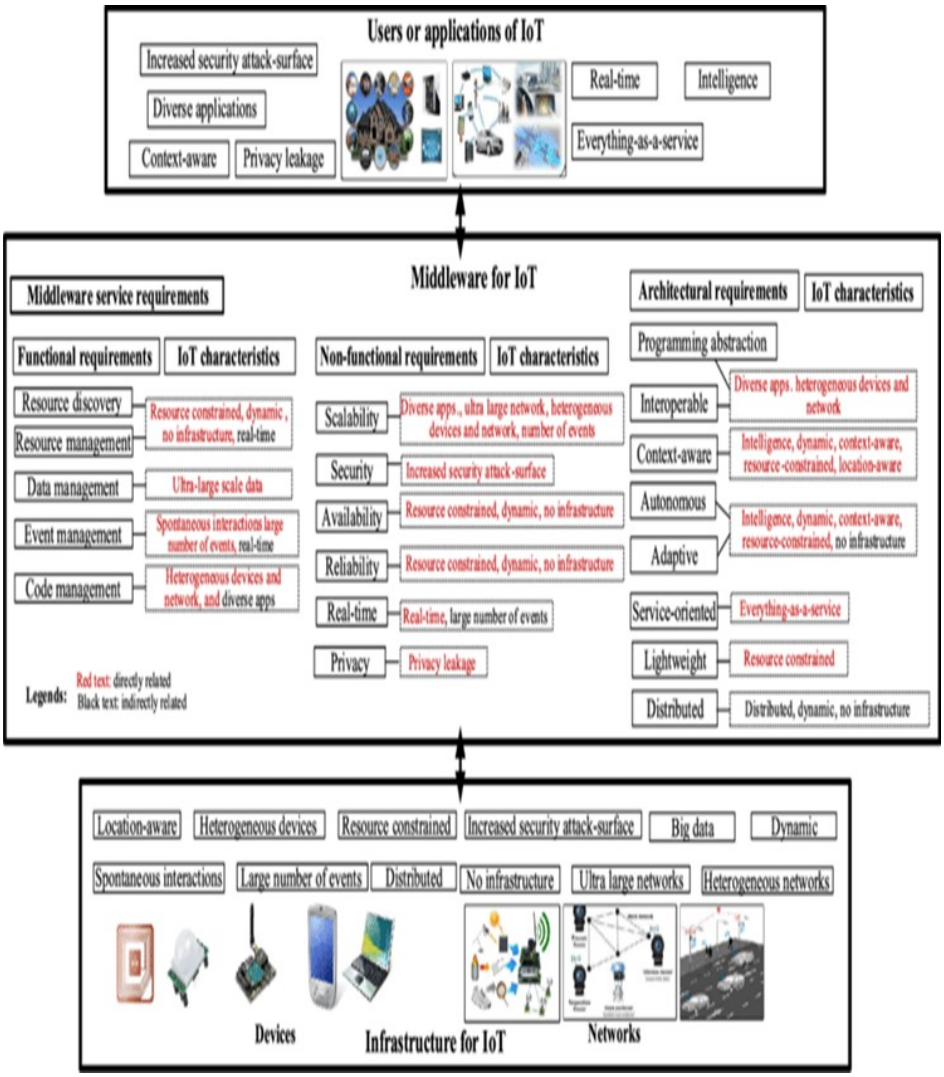



FIGURE 1.8 – Hétérogénéité des dispositifs

### 1.4.2 Ressources limitées

Les ressources limitées dans les systèmes IoT se réfèrent aux contraintes en matière de puissance de calcul, de mémoire, de bande passante et de batterie des dispositifs connectés. De nombreux appareils IoT sont conçus pour être petits, légers et économes en énergie, ce qui limite leur capacité à traiter de grandes quantités de données ou à exécuter des applications complexes (Shi et al., 2016) **1**. Cette contrainte impose des choix techniques, comme le besoin d'optimiser le traitement des données localement (edge computing) afin de réduire la dépendance au cloud et limiter l'utilisation du réseau. Les batteries des dispositifs IoT doivent être durables, mais elles ont une capacité limitée, ce qui nécessite des solutions d'optimisation énergétique pour prolonger leur autonomie. Par ailleurs, la bande passante reste souvent insuffisante pour transmettre des volumes massifs de données, obligeant à utiliser des stratégies de compression ou à traiter les données en temps réel pour réduire le besoin de communication (Gai et al., 2020) **4**. Ces limitations techniques exigent des compromis pour garantir des performances adéquates tout en respectant les ressources disponibles.

### 1.4.3 Mobilité

De plus en plus fréquemment, dans l'IoT, les dispositifs intelligents sont désormais capables de se déplacer. Parmi ces objets, on trouve des drones, des robots aspirateurs, ainsi que des véhicules connectés (Zhang et al., 2010) **1**. Afin d'assurer un accès permanent et transparent aux données qu'ils génèrent, mais aussi pour leur permettre d'interagir avec d'autres services IoT, ces dispositifs en mouvement (qu'il soit permanent ou ponctuel) introduisent de la dynamique dans le réseau, entraînant de multiples connexions et déconnexions en temps réel. Cela peut concerner des réseaux dynamiques formés temporairement par plusieurs dispositifs ou des réseaux statiques préexistants auxquels les dispositifs se connectent le temps nécessaire. La gestion de tels réseaux dynamiques devient particulièrement complexe dans de grands environnements distribués, car déterminer si un dispositif peut être accepté dans le réseau revient à évaluer le niveau de confiance qu'on lui accorde. Il s'agit donc, en fin de compte, d'une problématique de sécurité (Shi et al., 2016) **3**.

Dans les réseaux dynamiques, la gestion de la mobilité peut se faire de différentes manières. Le dispositif intelligent (qui, rappelons-le, est intégré à un objet physique) doit détecter son mouvement afin de savoir s'il va quitter sa position actuelle dans la topologie du réseau pour se connecter à un autre emplacement, voire à un autre réseau. Cette détection peut être réalisée par un scan passif des messages des participants au réseau ou par le suivi des émissions de balises. Une autre approche consiste à intégrer des mécanismes de signalisation et de contrôle de la localisation des nœuds directement dans les protocoles de communication du réseau. Il convient de noter que la mobilité augmente la surface d'attaque d'un réseau, et qu'elle va donc de pair avec la question de la sécurité (Gai et al., 2020) **5**. C'est pourquoi toute solution de sécurité pour les dispositifs IoT doit nécessairement prendre en compte leur mobilité.

### 1.4.4 Sécurité

La sécurité dans l'Internet des Objets (IoT) est un sujet essentiel et complexe, car elle touche à la protection des données, des dispositifs et des réseaux. L'IoT connecte une multitude de dispositifs intelligents et interconnectés, qui collectent, partagent et traitent des informations sensibles dans divers domaines tels que la santé, l'industrie, la domotique (Gubbi et al., 2013) **8**, etc. Ces dispositifs sont souvent déployés dans des environnements ouverts et peu surveillés, ce qui les rend vulnérables à une variété de menaces. Voici un aperçu des principaux enjeux de la sécurité dans l'IoT :

Les objets IoT génèrent une grande quantité de données, souvent sensibles (par exemple, des données de santé, des informations personnelles ou des paramètres industriels). La confidentialité de ces données est primordiale. Il est essentiel de mettre en place des mécanismes de chiffrement des communications pour éviter les interceptions ou les fuites de données. De plus, la gestion des accès doit être rigoureuse, pour s'assurer que seules les parties autorisées peuvent accéder aux informations collectées.

Les objets IoT, qui peuvent être des capteurs, des caméras, des véhicules ou des appareils médicaux, sont des cibles potentielles pour les attaquants. Un dispositif peut être cloné, modifié ou piraté pour compromettre son fonctionnement, corrompre ses données ou l'utiliser comme point d'entrée dans le réseau. L'authentification des dispositifs (par exemple, l'utilisation de certificats ou de clés privées) est une méthode courante pour garantir qu'un appareil est bien celui qu'il prétend être.

Les objets IoT sont souvent contraints par des ressources limitées, telles que la

puissance de calcul, la mémoire et la bande passante (Gai et al., 2020) **5**. Cela complique l'implémentation de protocoles de sécurité complexes, tels que des systèmes de chiffrement avancés ou des mécanismes d'authentification robustes. Il est donc nécessaire de concevoir des solutions de sécurité optimisées, adaptées aux capacités des dispositifs IoT tout en assurant un niveau de protection suffisant.

Les dispositifs IoT peuvent comporter des vulnérabilités logicielles ou matérielles qui, si elles ne sont pas corrigées rapidement, peuvent être exploitées par des attaquants. De plus, la gestion des mises à jour de sécurité est un défi majeur, car de nombreux objets IoT ne sont pas régulièrement mis à jour, soit en raison de limitations techniques, soit par négligence des utilisateurs. Les mises à jour doivent être sécurisées et faciles à déployer pour éviter que des failles ne demeurent ouvertes.

La sécurisation du réseau est un autre aspect critique, car les dispositifs IoT communiquent généralement via des réseaux sans fil (Wi-Fi, Bluetooth, Zigbee, etc.), qui peuvent être plus vulnérables aux attaques que les réseaux filaires (Roman et al., 2011) **1**. Il est donc important d'utiliser des protocoles de sécurité pour protéger les échanges de données, tels que le chiffrement des connexions et l'utilisation de VPN (réseaux privés virtuels) pour sécuriser les communications.

Dans des domaines sensibles, comme la santé, la sécurité des données est d'autant plus critique (Gai et al., 2020) **4**. Par exemple, dans les systèmes de télémédecine ou les dispositifs de surveillance médicale (comme les capteurs de glucose ou les pacemakers), les informations doivent être protégées contre tout accès non autorisé pour garantir la confidentialité des patients et la sécurité des traitements.

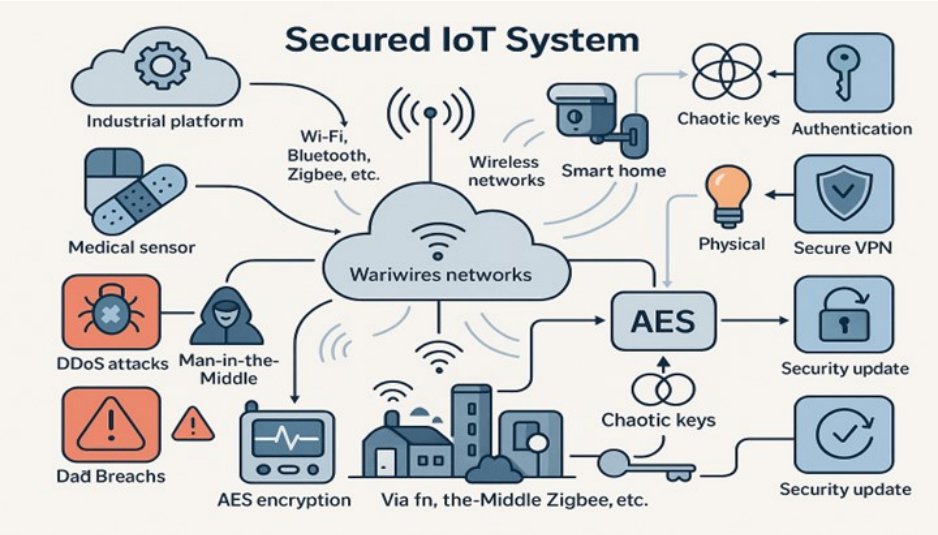



FIGURE 1.9 – Sécurité des IoT

## 1.5 L'IoT dans le domaine médical

L'Internet des Objets (IoT) transforme le domaine médical en offrant des solutions innovantes pour le suivi des patients et la gestion de la santé (Gubbi et al., 2013) **1**. Des dispositifs connectés, tels que les capteurs de pression artérielle, les moniteurs de glycémie ou les bracelets de suivi de l'activité physique, permettent de collecter des données en temps réel sur l'état de santé des patients. Ces données sont ensuite transmises aux professionnels de santé pour un suivi à distance, facilitant ainsi le diagnostic et

les décisions médicales (Chan, Campo, Estève, Fourniols, Escriba  Campo, 2008) **6 2**. L'IoT améliore également l'autonomie des personnes âgées ou des patients atteints de maladies chroniques en leur offrant une surveillance continue et personnalisée. De plus, il facilite la gestion des médicaments, comme les pompes à perfusion intelligentes, pour garantir un traitement optimal (Balta-Ozkan et al., 2013) **3**. Cependant, la sécurité et la confidentialité des données restent des enjeux majeurs pour protéger la vie privée des patients et garantir la fiabilité des dispositifs.

FIGURE 1.10 – L'IoT dans le domaine médical

## 1.6 Santé et assistance à l'autonomie à domicile

De nos jours, de plus en plus de personnes âgées tentent de vivre de manière autonome dans leur propre maison. Cependant, avec l'âge, rester en sécurité et indépendant est un défi majeur (Chan, Campo, Estève, Fourniols, Escriba  Campo, 2008) **6 1**. L'assistance à la vie ambiante a parmi ses principaux objectifs d'aider ces personnes âgées à atteindre cette autonomie. Elle vise aussi à assister les personnes à activités réduites (handicapées – par exemple les personnes en fauteuil roulant) ou, en général, les personnes qui ont simplement besoin d'une aide supplémentaire pour vivre seules. Les systèmes d'AAD peuvent aider les personnes à ne pas oublier leurs médicaments, à surveiller leurs modes de vie et leur santé, mais également à faire en sorte qu'elles se sentent moins isolées en proposant des services sociaux et des divertissements (Kim et al., 2020) **2**. L'IoT se présente donc comme une solution intéressante pour y parvenir.L'assistance à l'autonomie dans les maisons intelligentes L'assistance à la vie ambiante appliquée dans les maisons intelligentes bénéficie de différentes technologies de l'IoT, comme les capteurs ambiants, la reconnaissance visuelle, les systèmes de sécurité (détection de chute, vidéo surveillance) ce qui permet la surveillance de l'état de santé et des activités des personnes à travers plusieurs services comme l'illustre AAL Maison Intelligente Inclusion Supervision Prévention Détection Services de l'assistance à l'autonomie dans un domicile intelligent

FIGURE 1.11

### 1.6.1 L'assistance à l'autonomie dans les maisons intelligentes

L'assistance à l'autonomie dans les maisons intelligentes, souvent désignée par le terme AAL (Ambient Assisted Living), repose sur l'intégration de technologies issues de l'Internet des Objets (IoT) pour favoriser le maintien à domicile des personnes âgées, en situation de handicap ou ayant des besoins spécifiques d'assistance (Chan, Campo, Estève, Fourniols, Escriba  Campo, 2008) **6 1**. Ces maisons intelligentes sont équipées de capteurs, d'actionneurs, de caméras, d'assistants vocaux et de systèmes d'analyse permettant de surveiller en continu l'état de santé, les activités et les habitudes des résidents.

#### Objectifs principaux de l'AAL

- Prévenir les accidents (ex. : détection de chutes, alertes d'inactivité inhabituelle),
- Surveiller la santé (ex. : suivi des constantes vitales, rappels de prise de médicaments),
- Renforcer la sécurité (ex. : surveillance vidéo, détecteurs de fumée, fermeture automatique des portes),
- Soutenir le lien social (ex. : communication facile avec les proches et les soignants),
- Favoriser l'autonomie (ex. : assistance vocale, automatisation des tâches domestiques).

#### Technologies mises en œuvre

Les maisons intelligentes combinent plusieurs technologies IoT :

- Capteurs ambiants : détectent les mouvements, les chutes, la température ou la qualité de l'air.
- Reconnaissance visuelle : permet d'identifier les comportements à risque ou les situations anormales.
- Automatisation : contrôle des équipements (éclairage, chauffage, volets) selon les habitudes de l'utilisateur.
- Systèmes de communication : notification automatique aux aidants ou aux services d'urgence.

**Avantages**

- Préservation de l'autonomie : les résidents peuvent vivre plus longtemps chez eux en toute sécurité.
- Réduction des hospitalisations : grâce à la détection précoce de problèmes de santé.
- Soulagement des aidants : qui peuvent surveiller l'état de leurs proches à distance.
- Amélioration de la qualité de vie : en apportant confort, sécurité et confiance.

**Enjeux**

Malgré ses nombreux avantages, l'assistance à l'autonomie soulève également des questions éthiques et sécuritaires : respect de la vie privée, sécurisation des données personnelles collectées, consentement des utilisateurs et accessibilité financière de ces technologies.

## 1.7 Principaux Concepts de Base en Sécurité dans l'Internet des Objets

La sécurité dans l'Internet des Objets (IoT) est cruciale pour assurer la protection des données, des dispositifs et des réseaux interconnectés. Les objets IoT, souvent déployés dans des environnements ouverts et vulnérables, nécessitent des mécanismes de sécurité robustes pour prévenir les attaques physiques, logicielles ou par réseau (Roman et al., 2011) **1**. La confidentialité des données collectées et transmises par ces objets, ainsi que l'intégrité de ces données, sont des priorités pour garantir qu'elles ne soient ni altérées ni interceptées (Shi et al., 2016) **2**. De plus, l'authentification des dispositifs et des utilisateurs est essentielle pour s'assurer que seuls les acteurs légitimes peuvent accéder aux systèmes. La gestion des risques inclut également la mise à jour régulière des systèmes, notamment des logiciels et du firmware des objets, pour corriger les vulnérabilités potentielles. La sécurité des communications entre les dispositifs IoT, ainsi que le contrôle des accès, garantit que les informations sensibles sont protégées contre les intrusions. Ces mesures sont particulièrement importantes dans des secteurs tels que la santé, où des données sensibles sont souvent collectées, et dans des applications industrielles où la fiabilité des dispositifs est primordiale.

### 1.7.1 Confidentialité

La confidentialité assure que seules les parties autorisées peuvent accéder aux informations sensibles transmises ou stockées par les dispositifs IoT. Cela inclut l'utilisation du chiffrement des données, tant au repos qu'en transit, pour empêcher l'accès non autorisé (Shi et al., 2016) **2**.

### 1.7.2 Intégrité

L'intégrité des données garantit que les informations transmises entre les dispositifs IoT ne sont pas modifiées ou altérées de manière non autorisée. Des mécanismes comme les fonctions de hachage et les signatures numériques sont utilisés pour vérifier que les données restent intactes tout au long de leur parcours.

### 1.7.3 Disponibilité

La disponibilité assure que les dispositifs IoT et les données sont accessibles en temps voulu et sans interruption. Les attaques par déni de service (DoS/DDoS) peuvent cibler cette disponibilité en inondant un réseau de trafic malveillant, rendant les systèmes inaccessibles.

### 1.7.4 Identification et Authentification

L'identification et l'authentification dans l'IoT assurent que seuls les dispositifs et utilisateurs légitimes accèdent aux systèmes et aux données. L'identification consiste à vérifier l'entité en question, tandis que l'authentification valide son identité via des méthodes comme les certificats numériques, les clés cryptographiques ou l'authentification multi-facteurs. Les protocoles comme OAuth ou X.509 permettent de sécuriser ces processus. Ces mécanismes sont essentiels pour prévenir l'accès non autorisé et garantir la confidentialité des informations sensibles. Dans des environnements comme la santé ou l'industrie, une authentification forte est cruciale (Abdmeziem, 2016) **2**.

### 1.7.5 Non-Répudiation

La non-répudiation est la garantie qu'une action a été effectuée par une partie. En particulier dans le contexte des communications, la non-répudiation est l'assurance (c'est-à-dire la preuve) qu'un émetteur ne peut nier avoir transmis des messages ; on parle de preuve de l'origine. Là, aussi, c'est souvent le mécanisme cryptographique de signature numérique qui est utilisé en pratique pour assurer cette propriété (JENOVA, 2023) **2**.

### 1.7.6 Contrôle d'accès

Le contrôle d'accès dans l'IoT permet de réguler qui peut accéder à quels dispositifs et données au sein d'un réseau IoT. Il repose sur des mécanismes comme l'authentification, la gestion des rôles et des permissions pour limiter l'accès aux ressources sensibles. Des systèmes de contrôle d'accès basés sur des rôles (RBAC) ou des attributs (ABAC) sont couramment utilisés pour définir et appliquer des politiques d'accès. Ces mécanismes sont essentiels pour protéger les objets IoT et les informations sensibles contre les accès non autorisés (Aliane, 2020) **6**.

## 1.8 Principales Attaques Visant Les Objets Connectés

Les objets connectés (IoT) sont vulnérables à diverses attaques, en raison de leurs ressources limitées, de leur interconnexion et de leur déploiement souvent dans des environnements ouverts. Voici les principales attaques visant les objets connectés :

1. **Attaque par déni de service (DoS/DDoS)** : Les dispositifs IoT peuvent être utilisés comme vecteurs d'attaques par déni de service, en inondant le réseau de trafic malveillant, ce qui rend les services indisponibles. Cela peut être particulièrement dangereux lorsque de nombreux appareils sont compromis pour créer un botnet, comme lors de l'attaque Mirai (TechNews, 2025) **2**.

2. **Interception et écoute (Eavesdropping)** : L'attaque d'interception consiste à écouter les communications entre les dispositifs IoT pour collecter des informations sensibles. Le manque de chiffrement peut permettre à des attaquants d'intercepter des données comme des informations personnelles ou des clés de sécurité (CNRS, 2023) **3**.
3. **Injection de code malveillant** : Les attaquants peuvent exploiter des vulnérabilités dans le firmware des objets connectés pour injecter du code malveillant. Cela peut permettre de prendre le contrôle du dispositif, de modifier ses fonctions ou de l'utiliser à des fins malveillantes (Ameen Abu-Sharkh, 2023) **4**.
4. **Clonage et usurpation d'identité** : L'attaque par clonage consiste à copier l'identité d'un dispositif IoT légitime (comme son adresse MAC ou son certificat) pour l'utiliser à des fins malveillantes, contournant ainsi les mécanismes d'authentification (Zscaler, 2025) **5**.
5. **Attaque de type "man-in-the-middle" (MITM)** : Dans cette attaque, l'attaquant intercepte les communications entre deux dispositifs IoT pour modifier, voler ou insérer des informations sans que les parties en soient conscientes. Cette attaque est particulièrement risquée si la communication n'est pas chiffrée (Vaadata, 2024) **6**.
6. **Exploitation des vulnérabilités physiques** : En raison de leur déploiement souvent dans des environnements ouverts, les dispositifs IoT peuvent être attaqués physiquement (par exemple, par débranchement, sabotage ou modification directe de leurs composants) (Vaadata, 2024) **6**.

Ces attaques soulignent l'importance d'une sécurité robuste dans la conception, le déploiement et la gestion des objets connectés. La mise en œuvre de mécanismes de protection, comme le chiffrement, l'authentification forte et la mise à jour régulière des dispositifs, est essentielle pour minimiser ces risques.

## 1.9 Solution et mécanisme de sécurité pour l'internet des objets

Pour sécuriser l'Internet des Objets (IoT) face aux menaces et aux vulnérabilités potentielles, il est essentiel d'adopter des solutions et mécanismes de sécurité adaptés (Shunlongwei, 2024) **5**. Voici quelques approches clés pour protéger les objets connectés, leurs données et leur réseau :

1. **Chiffrement léger des données dans les systèmes IoT**
   Le chiffrement est l'une des solutions les plus efficaces pour sécuriser les échanges de données entre les dispositifs IoT et les serveurs. Dans un contexte où les ressources sont souvent limitées (batterie, puissance de calcul), il est crucial d'utiliser des algorithmes optimisés pour la rapidité et l'efficacité (Lara et al., 2018) **17**.

   Parmi les méthodes de chiffrement léger, on retrouve :

   - **Salsa20** : Algorithme de chiffrement en flux rapide et sécurisé, conçu pour offrir une excellente performance sur des architectures à faible puissance, idéal pour les communications IoT.

- **ChaCha20** : Variante améliorée de Salsa20, elle garantit une sécurité renforcée tout en conservant une grande vitesse de chiffrement, adaptée aux flux de données en temps réel.

- **XTEA (Extended Tiny Encryption Algorithm)** : Algorithme symétrique simple et efficace, consommant peu de ressources, parfait pour les dispositifs IoT contraints.

- **LEA (Lightweight Encryption Algorithm)** : Conçu spécialement pour les environnements embarqués, il offre un bon compromis entre sécurité et rapidité.

- **Speck & Simon** : Algorithmes de chiffrement développés pour les systèmes à faible puissance, largement utilisés dans les capteurs et les objets connectés.

En complément, les protocoles TLS/SSL restent essentiels pour garantir une transmission sécurisée des données, et les solutions VPN permettent de créer des tunnels chiffrés sur des réseaux non sécurisés.

Le choix du chiffrement dépend des contraintes spécifiques des systèmes IoT : optimisation de la consommation énergétique, vitesse d'exécution et résistance aux attaques. L'utilisation d'algorithmes adaptés assure une sécurité renforcée sans impacter la performance globale du système.

2. **Authentification et gestion des identités**
   L'authentification et la gestion des identités sont cruciales pour garantir que seuls les dispositifs et utilisateurs autorisés peuvent accéder aux ressources du réseau IoT. Cela inclut l'utilisation de certificats numériques, de clés cryptographiques et de mécanismes d'authentification forte, comme l'authentification multi-facteurs (MFA). Les protocoles comme OAuth 2.0 et X.509 peuvent être utilisés pour sécuriser l'accès aux ressources (Keyfactor, 2020) **11**.
3. **Contrôle d'accès basé sur les rôles (RBAC)**
   Le contrôle d'accès permet de gérer qui peut accéder à quoi dans un environnement IoT. Le contrôle d'accès basé sur des rôles (RBAC) ou sur des attributs (ABAC) permet de définir des permissions d'accès en fonction du rôle ou des attributs d'un utilisateur ou d'un dispositif. Cela limite l'accès aux données et aux systèmes IoT à ceux qui en ont besoin pour fonctionner (Okta, 2024) **2**.
4. **Mises à jour de sécurité et gestion des vulnérabilités**
   La gestion des vulnérabilités implique de surveiller et de corriger régulièrement les failles de sécurité dans le firmware et le logiciel des dispositifs IoT. Les mises à jour automatiques de sécurité sont essentielles pour réduire les risques liés aux vulnérabilités connues. Cela comprend également le déploiement de correctifs de sécurité pour garantir la protection continue des dispositifs connectés (Shunlongwei, 2024) **5**.
5. **Détection des intrusions et surveillance continue**
   Les systèmes de détection des intrusions (IDS/IPS) permettent de surveiller en temps réel le trafic réseau et les comportements des dispositifs IoT pour détecter des activités anormales ou suspectes. En cas d'intrusion, ces systèmes peuvent alerter les administrateurs et déclencher des actions correctives pour limiter les dommages (Khernane, 2025) **32**.
6. **Isolation et segmentation des réseaux**
   La segmentation du réseau permet de diviser un réseau IoT en sous-réseaux isolés afin de limiter la propagation des attaques. Par exemple, les dispositifs

critiques peuvent être isolés du reste du réseau, réduisant ainsi le risque qu'une attaque sur un appareil IoT affecte d'autres parties du réseau.

7. **Blockchain pour la gestion des identités et des transactions**
   La blockchain peut être utilisée pour garantir la sécurité des transactions et la gestion des identités dans les réseaux IoT. Elle offre une méthode décentralisée et inviolable de suivre les interactions entre les dispositifs IoT, ce qui renforce la sécurité et la traçabilité des données.
8. **Sécurisation des dispositifs physiques**
   Les dispositifs IoT peuvent être protégés physiquement contre le vol, la manipulation ou l'accès non autorisé. Cela inclut l'utilisation de boîtiers sécurisés, de verrouillages physiques et d'autres mécanismes de protection pour empêcher l'accès direct aux composants matériels sensibles.
9. **Virtualisation et conteneurisation**
   Les technologies de virtualisation et de conteneurisation permettent de séparer les processus et les fonctions des objets IoT dans des environnements virtuels sécurisés. Cela permet d'éviter qu'un compromis dans un composant n'affecte l'ensemble du système IoT.
10. **Politiques de sécurité et éducation des utilisateurs**
    Enfin, l'adoption de politiques de sécurité claires, accompagnées de bonnes pratiques, est essentielle pour garantir la sécurité dans un environnement IoT. Cela inclut la formation des utilisateurs et des administrateurs sur les bonnes pratiques de sécurité, comme la gestion des mots de passe, l'activation des mises à jour automatiques et la gestion des vulnérabilités (Internet Society, 2019) **26**.

Ces solutions et mécanismes doivent être combinés pour créer une architecture IoT sécurisée, protégeant non seulement les dispositifs, mais aussi les données sensibles et les utilisateurs finaux contre les menaces potentielles. La mise en œuvre proactive de la sécurité, avec une surveillance continue et une mise à jour des mesures, est essentielle pour faire face à l'évolution constante des cybermenaces dans l'IoT.

## 1.10 Conclusion

L'Internet des Objets (IoT) se caractérise par l'interconnexion massive de dispositifs intelligents, facilitant ainsi l'échange de données et l'automatisation dans des domaines variés tels que la santé, l'industrie et les maisons intelligentes. Toutefois, cet environnement dynamique présente d'importantes vulnérabilités en matière de sécurité. La diversité des objets connectés, leurs ressources limitées et la multiplicité des protocoles de communication amplifient les risques liés aux cyberattaques et aux violations de données (Saleh et al., 2024) **2**.

Dans ce contexte, la cryptographie apparaît comme un rempart essentiel pour assurer la confidentialité, l'intégrité et l'authenticité des échanges. Parmi les solutions les plus robustes, l'Advanced Encryption Standard (AES) s'est imposé comme une référence grâce à sa structure fondée sur des transformations mathématiques avancées et la possibilité de choisir parmi plusieurs tailles de clés (128, 192 ou 256 bits) (Bajpeyi Verma, 2024) **3**. Il est largement utilisé afin de sécuriser les communications ainsi que le stockage des données.

Cependant, l'implémentation de l'AES dans un environnement IoT soulève des défis spécifiques. Les contraintes matérielles des objets connectés, la nécessité d'une exécution

rapide et la gestion sécurisée des clés imposent de repenser l'algorithme classique pour qu'il soit réellement adapté à ces environnements contraints. C'est dans ce cadre que s'inscrit l'objectif de ce travail.

Mes intentions et objectifs pour la suite sont les suivants :

- Optimiser AES pour l'IoT : Adapter l'algorithme afin de répondre aux limitations en termes de puissance de calcul, de mémoire et d'énergie.

- Intégrer des mécanismes chaotiques : Exploiter la sensibilité aux conditions initiales des systèmes chaotiques – via des outils comme la carte logistique – pour générer dynamiquement des clés, des vecteurs d'initialisation et des S-box plus imprévisibles et sécurisés (El Gaabouri et al., 2024) **5**.

- Améliorer la gestion des clés : Proposer des solutions innovantes pour le stockage, la mise à jour et la distribution des clés dans un environnement distribué et potentiellement hostile.

- Valider l'approche par l'expérimentation : Mesurer les performances de l'AES modifié en termes de débit, de latence et d'efficience énergétique, afin de démontrer la faisabilité et l'efficacité de cette approche dans un contexte IoT réel (Farooq et al., 2020) **1**.

Ce chapitre a ainsi permis de dresser le bilan des enjeux sécuritaires dans l'IoT et d'identifier les limites de l'implémentation classique d'AES. Les développements futurs s'orienteront vers la mise en œuvre d'un AES allégé et renforcé par des mécanismes chaotiques, afin de proposer une solution cryptographique innovante, adaptée aux contraintes des dispositifs IoT tout en offrant une protection optimale contre les cybermenaces.

# Chapitre 2

# Chiffrement AES

## 2.1 Introduction

L'Advanced Encryption Standard (AES) est un algorithme de chiffrement symétrique largement utilisé pour garantir la confidentialité des données. Il repose sur un ensemble d'opérations bien définies appliquées sur des blocs de 128 bits avec des clés de 128, 192 ou 256 bits **Veritas2024**. AES est reconnu pour sa robustesse cryptographique et ses performances dans des environnements standards.

Cependant, son intégration dans les environnements IoT (Internet of Things) soulève plusieurs défis. En effet, les objets connectés possèdent souvent des ressources limitées en termes de puissance de calcul, de mémoire, de bande passante et d'énergie. AES, bien que efficace sur des systèmes classiques, peut s'avérer trop coûteux pour certains capteurs ou dispositifs embarqués. De plus, le coût d'exécution de l'expansion de clé et des opérations sur plusieurs tours ralentit le traitement en temps réel. Ainsi, bien que sécurisé, AES dans sa forme classique n'est pas toujours adapté aux contraintes des environnements IoT, ce qui a motivé le développement de versions allégées ou modifiées de l'algorithme **Bajpeyi2024**.

### 2.1.1 Qu'est-Ce Que Le Chiffrement AES ?

Le chiffrement AES est un algorithme de chiffrement symétrique, ce qui signifie qu'il utilise la même clé pour le chiffrement et le déchiffrement des données. Il a été adopté comme standard de chiffrement par le gouvernement des États-Unis en 2001, après un processus rigoureux de sélection parmi plusieurs candidats pour remplacer le DES (Data Encryption Standard), qui était devenu vulnérable aux attaques modernes **PandaSecurity2024**.

L'AES fonctionne sur des blocs de données de taille fixe, typiquement 128 bits, et peut être utilisé avec des clés de tailles différentes : 128, 192 ou 256 bits. En raison de sa robustesse et de son efficacité, AES est devenu le standard de facto pour le chiffrement des données sensibles à travers le monde, qu'il s'agisse de communications sécurisées, de stockage de données ou de protection des transactions financières **Fouque2023**.

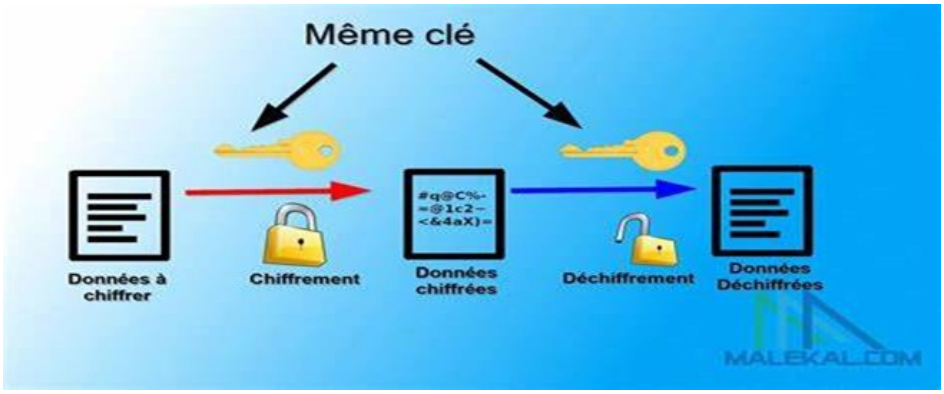



FIGURE 2.1 – Chiffrement AES

### 2.1.2 Historique Du Chiffrement AES

L'origine du chiffrement AES remonte à la fin des années 1990, lorsque le National Institute of Standards and Technology (NIST) des États-Unis a lancé un appel à propositions pour remplacer le DES. Le DES, bien qu'efficace dans les années 1970,

était devenu vulnérable face à l'augmentation de la puissance de calcul des ordinateurs modernes. En 1997, NIST a ouvert un concours international pour sélectionner un nouvel algorithme de chiffrement symétrique qui répondrait aux exigences de sécurité, de performance et de flexibilité pour une utilisation à long terme **NIST2001**.

Le concours a attiré des propositions de nombreux chercheurs en cryptographie, et après plusieurs années d'examen, le candidat Rijndael, développé par les cryptographes belges Vincent Rijmen et Joan Daemen, a été retenu en 2000. En 2001, le NIST a officiellement adopté l'algorithme Rijndael comme le standard AES, en raison de sa sécurité, de sa rapidité et de sa capacité à s'adapter à des tailles de clés variées (128, 192 et 256 bits).

### 2.1.3 Objectifs Et Applications Du Chiffrement AES

L'objectif principal du chiffrement AES est de protéger la confidentialité des données contre tout accès non autorisé, tout en assurant l'intégrité des informations transmises ou stockées **W3rOne2024**. Grâce à sa robustesse et à sa flexibilité, AES est utilisé dans une large gamme d'applications de sécurité, telles que :

- Sécurisation des communications : AES est largement utilisé dans les protocoles de communication sécurisés comme SSL/TLS, qui protègent les échanges de données sensibles sur Internet (par exemple, les transactions bancaires en ligne).

- Stockage de données : De nombreuses applications de stockage, comme les disques durs chiffrés ou les services de cloud, utilisent AES pour protéger les informations des utilisateurs contre les accès non autorisés.

- Systèmes de paiement : AES est utilisé pour sécuriser les transactions financières dans les systèmes de cartes bancaires, les paiements mobiles, et les solutions de portefeuille électronique **Editverse2022**.

- Applications militaires et gouvernementales : L'AES est largement déployé dans les agences gouvernementales et militaires pour la protection des données sensibles et confidentielles **W3rOne2024**.

- Cryptomonnaies et Blockchain : Les systèmes de blockchain, qui sous-tendent les cryptomonnaies, emploient souvent AES pour assurer la sécurité des données **Springer2022**.

En résumé, le chiffrement AES est une pierre angulaire de la sécurité moderne, offrant une protection de haute qualité pour une grande variété d'applications, tout en restant flexible et performant pour faire face aux besoins croissants de sécurité dans un monde numérique de plus en plus interconnecté.

## 2.2 Fondements Théoriques Du Chiffrement

Le chiffrement est une technique qui transforme des informations lisibles en données inintelligibles, dans le but de préserver la confidentialité et l'intégrité de l'information. Cette section explore les concepts fondamentaux du chiffrement, les différents types d'algorithmes de chiffrement symétriques, et la comparaison entre AES et d'autres algorithmes populaires.

### 2.2.1 Concepts De Base Du Chiffrement

Le chiffrement repose sur un ensemble de concepts et de processus clés permettant de transformer des données en une forme protégée contre l'accès non autorisé. Voici les concepts essentiels du chiffrement :

#### Texte en clair et texte chiffré

- Le texte en clair (*plaintext*) est le message original que l'on souhaite protéger.
- Le texte chiffré (*ciphertext*) est la version modifiée du texte en clair, obtenue après application d'un algorithme de chiffrement.

#### Clé de chiffrement

La clé est un élément crucial dans le chiffrement. C'est une valeur qui est utilisée par l'algorithme de chiffrement pour transformer le texte en clair en texte chiffré.

Selon le type d'algorithme, la clé peut être partagée entre les utilisateurs (dans le cas du chiffrement symétrique) ou publique/privée (dans le cas du chiffrement asymétrique).

#### Algorithmes de chiffrement

Un algorithme de chiffrement définit les règles ou les étapes utilisées pour transformer le texte en clair en texte chiffré (et vice versa, dans le cas du déchiffrement). L'algorithme peut être soit symétrique, soit asymétrique.

#### Algorithmes de Déchiffrement

Le déchiffrement est le processus inverse du chiffrement, où le texte chiffré est converti en texte en clair en utilisant une clé correspondante. Dans le chiffrement symétrique, la même clé est utilisée pour chiffrer et déchiffrer, tandis que dans le chiffrement asymétrique, des clés différentes sont utilisées pour le chiffrement et le déchiffrement.

#### Sécurité du chiffrement

Un algorithme de chiffrement est considéré comme sécurisé s'il résiste aux attaques pour découvrir la clé ou déchiffrer les données sans la clé correcte. L'attaque la plus courante contre un algorithme est l'attaque par force brute, qui consiste à essayer toutes les clés possibles jusqu'à ce que le texte chiffré soit déchiffré avec succès.

### 2.2.2 Les Algorithmes De Chiffrement Symétriques

Les algorithmes de chiffrement symétriques sont des algorithmes dans lesquels la même clé est utilisée à la fois pour le chiffrement et le déchiffrement. Ils sont largement utilisés pour leur rapidité et leur efficacité, en particulier lorsque de grandes quantités de données doivent être protégées **Smith2020**.

#### Caractéristiques principales des algorithmes symétriques

- Clé unique : La même clé est partagée entre les parties pour chiffrer et déchiffrer.
- Vitesse : Les algorithmes symétriques sont généralement plus rapides que les algorithmes asymétriques.
- Sécurité : La sécurité repose sur la confidentialité de la clé. Si la clé est compromise, l'ensemble du système de chiffrement est vulnérable.

#### Exemples d'algorithmes symétriques

- DES (Data Encryption Standard) : Un ancien standard de chiffrement symétrique, qui a été largement utilisé avant d'être remplacé par AES en raison de sa faible sécurité (clé de 56 bits).
- 3DES (Triple DES) : Une version améliorée de DES, qui applique trois fois le chiffrement DES avec trois clés différentes. Cependant, 3DES est désormais considéré comme obsolète en raison de sa lenteur et de sa sécurité limitée.
- AES (Advanced Encryption Standard) : L'algorithme symétrique le plus populaire actuellement, utilisé pour sécuriser une large gamme d'applications.

#### Modes de fonctionnement des algorithmes symétriques

- Mode ECB (Electronic Codebook) : Chaque bloc de texte en clair est chiffré indépendamment. Bien que simple, ce mode est vulnérable à des attaques par analyse de fréquence.
- Mode CBC (Cipher Block Chaining) : Chaque bloc de texte en clair est XORé avec le bloc précédent avant d'être chiffré, ce qui améliore la sécurité.
- Mode CTR (Counter) : Utilise un compteur pour générer un flux de clé, ce qui permet de chiffrer des données de longueur variable.

### 2.2.3 Comparaison Entre AES Et D'autres Algorithmes De Chiffrement

AES est l'un des algorithmes symétriques les plus utilisés aujourd'hui, mais il existe d'autres algorithmes de chiffrement, chacun avec ses caractéristiques et son domaine d'application. Voici une comparaison entre AES et certains autres algorithmes populaires :

#### AES vs DES (Data Encryption Standard)

- Sécurité : AES offre une sécurité beaucoup plus élevée que DES. En effet, la clé de 128 bits d'AES est largement plus résistante aux attaques par force brute que la clé de 56 bits de DES **NIST2001**.
- Vitesse : AES est généralement plus rapide que DES, en particulier sur des architectures modernes.
- Utilisation : DES est obsolète et n'est plus utilisé dans les applications modernes, tandis que AES est largement adopté dans divers secteurs (gouvernement, finances, etc.).

**AES vs 3DES (Triple DES)**

- Sécurité : 3DES offre une sécurité supérieure à celle de DES, mais il est moins performant que AES, qui est conçu pour résister à des attaques modernes tout en étant plus rapide.

- Vitesse : AES est beaucoup plus rapide que 3DES, qui nécessite trois passes de chiffrement sur chaque bloc de données.

- Utilisation : 3DES est progressivement remplacé par AES en raison de sa lenteur et de sa vulnérabilité par rapport aux normes actuelles de sécurité.

**AES vs Blowfish**

- Sécurité : Blowfish est également un algorithme de chiffrement symétrique à clé secrète, mais il n'atteint pas le même niveau de sécurité que AES avec des tailles de clés plus longues **Schneier1993**.

- Vitesse : Blowfish est plus rapide que AES pour de petites tailles de clés, mais pour des clés de 256 bits, AES surpasse Blowfish.

- Utilisation : Blowfish est parfois utilisé dans des applications moins exigeantes en termes de sécurité et de performance, alors qu'AES est privilégié pour des systèmes nécessitant une sécurité robuste.

**AES vs ChaCha20**

- Sécurité : ChaCha20 est un algorithme de chiffrement symétrique basé sur un flux de chiffrement, utilisé dans certains cas comme alternative à AES. Il est réputé pour sa robustesse contre certaines attaques **Bernstein2008**.

- Vitesse : AES est plus rapide que ChaCha20 sur des appareils avec des accélérateurs matériels, mais ChaCha20 peut être plus efficace sur des appareils plus simples, comme les smartphones, où les opérations AES peuvent être plus coûteuses en termes de consommation d'énergie.

- Utilisation : ChaCha20 est utilisé dans des contextes où des performances élevées sont requises, comme dans le chiffrement des données réseau avec TLS ou dans les systèmes mobiles.

Cette section présente les concepts fondamentaux du chiffrement et donne une vue d'ensemble des algorithmes symétriques et de leur comparaison avec AES. Ces informations sont cruciales pour comprendre les avantages et les inconvénients de chaque méthode de chiffrement dans un contexte donné.

## 2.3 Architecture De L'algorithme AES

L'algorithme AES (Advanced Encryption Standard) repose sur une architecture solide et bien définie. Il utilise une approche de substitution-permutation pour effectuer le chiffrement, offrant à la fois sécurité et efficacité. Cette section présente la structure générale de l'AES, les tailles de clés, le modèle de réseau de substitution-permutation (SPN), et les différentes étapes du processus de chiffrement **DaemenRijmen2002**.

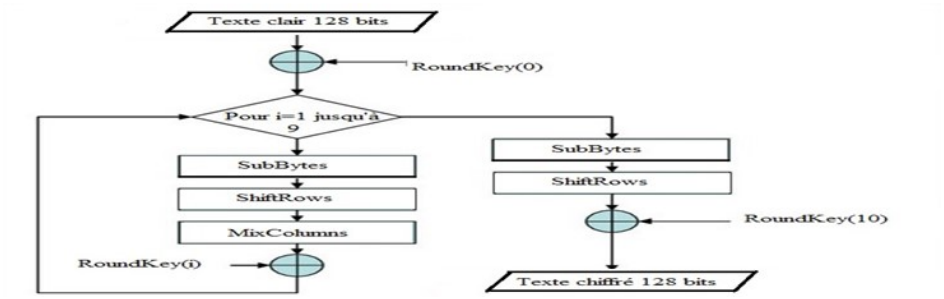



FIGURE 2.2 – Principes chiffrement AES

### 2.3.1 Structure Générale De L'AES

L'algorithme AES est basé sur une structure à bloc, ce qui signifie qu'il chiffre les données par blocs de taille fixe. Dans le cas de l'AES, chaque bloc de texte en clair a une taille de 128 bits (16 octets). L'AES peut être utilisé avec trois tailles de clés différentes : 128 bits, 192 bits, et 256 bits.

La structure de l'AES est organisée en étapes de transformation répétées au cours de plusieurs tours (*rounds*). Le nombre de tours dépend de la taille de la clé :

- 10 tours pour une clé de 128 bits,
- 12 tours pour une clé de 192 bits,
- 14 tours pour une clé de 256 bits.

Les transformations effectuées à chaque tour sont appliquées à un tableau appelé État (*State*). L'État est un tableau de $4 \times 4$ octets (un total de 16 octets, ou 128 bits) qui contient le texte en clair ou le texte chiffré, selon l'étape du chiffrement.

L'AES fonctionne selon un processus itératif qui applique des opérations de substitution, de transposition, et de fusion des données, assurant ainsi une diffusion et confusion efficaces des bits du texte en clair.

### 2.3.2 Les Tailles De Clés : 128, 192 Et 256 Bits

AES permet trois tailles de clés différentes, chacune ayant des propriétés spécifiques :

#### Clé de 128 bits (AES-128)

La clé de 128 bits est la plus couramment utilisée et présente un bon équilibre entre performance et sécurité. Elle génère 10 tours de chiffrement, ce qui est suffisant pour garantir une protection contre les attaques modernes.

#### Clé de 192 bits (AES-192)

AES-192 utilise une clé de 192 bits, offrant un niveau de sécurité plus élevé que l'AES-128. Cette taille de clé génère 12 tours de chiffrement, ce qui rend l'algorithme plus sécurisé, mais aussi plus lent en termes de performance.

#### Clé de 256 bits (AES-256)

AES-256 offre la plus grande sécurité, avec une clé de 256 bits, et génère 14 tours de chiffrement. Bien que cette taille de clé offre la meilleure sécurité, elle est également la plus lente en raison du nombre plus élevé de tours.

En fonction des besoins en matière de performance et de sécurité, AES permet de choisir la taille de clé la plus appropriée. AES-128 est largement utilisé dans de nombreuses applications, tandis que AES-256 est privilégié pour les environnements nécessitant une sécurité maximale.

### 2.3.3 Les Étapes De L'AES : SubBytes, ShiftRows, MixColumns Et AddRoundKey

Le chiffrement AES se compose de plusieurs étapes qui sont répétées au cours des tours. Ces étapes sont définies comme suit :

#### SubBytes

L'opération `SubBytes` effectue une substitution non linéaire sur chaque octet de l'État. Chaque octet est remplacé par un autre octet en fonction de la S-box, qui est une table de substitution fixe. Cela crée de la confusion, un élément clé de la cryptographie, car elle permet de masquer les relations directes entre le texte en clair et le texte chiffré.

#### ShiftRows

Après la substitution, l'opération `ShiftRows` effectue un décalage circulaire des lignes de l'État. La première ligne reste inchangée, la deuxième ligne est décalée d'un octet vers la gauche, la troisième ligne est décalée de deux octets, et la quatrième ligne est décalée de trois octets. Cette opération aide à mélanger les octets dans l'État et permet de mieux diffuser l'information.

#### MixColumns

L'opération `MixColumns` est une transformation de diffusion qui s'applique sur les colonnes de l'État. Elle mélange les octets de chaque colonne pour assurer que chaque octet de l'État dépend de tous les autres. Cela rend le chiffrement plus robuste contre certaines attaques cryptographiques.

#### AddRoundKey

Enfin, l'opération `AddRoundKey` est effectuée à chaque tour. L'État est combiné avec un sous-ensemble de la clé de chiffrement en utilisant une opération XOR. Cela assure que chaque tour de chiffrement dépend non seulement de l'État actuel mais aussi de la clé.

#### Tour Final

Lors du dernier tour (selon la taille de la clé), l'opération `MixColumns` est omise, car elle n'est pas nécessaire pour la dernière transformation. Les étapes restantes (`SubBytes`, `ShiftRows`, et `AddRoundKey`) sont appliquées, ce qui produit le texte chiffré final.

Ces étapes de transformation itératives et leur combinaison créent un algorithme de chiffrement puissant, capable de résister aux attaques modernes tout en étant relativement rapide dans son exécution.

## 2.4 Description de l'AES

### 2.4.1 Création de la matrice State

AES est un système de chiffrement itéré. Son principe est de répéter $N_e$ fois sur des blocs de 128 bits une fonction d'étage $\mathcal{J}_g$, où $N_e$ peut aller de 10 à 14. Le nombre d'itérations dépend de la taille choisie pour la clé.

Nous notons dans ce qui suit :

- $X$ : le bloc initial issu du texte clair codé en hexadécimal, il est de 128 bits.
- $W_i$ : le texte en entrée de l'étage $i$.
- *State* : correspond à l'écriture du flux d'entrée $W_i$ sous forme de matrice. Nous considérons pour cela le bloc de 128 bits $W_i$ comme une suite de 16 octets que l'on range en une matrice $4 \times 4$. Chaque case de cette matrice est alors un élément de taille 8 bits.
- $Y$ : le bloc crypté final.

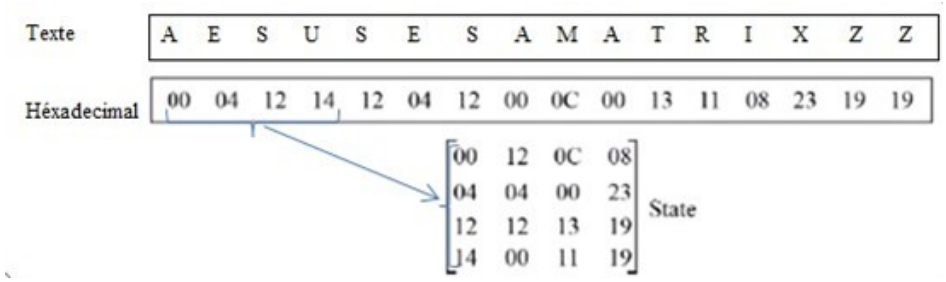



FIGURE 2.3 – Creation de la matrice

$\oplus$ est bien évidemment un XOR.

'$g$' peut être définie comme étant l'ensemble des transformations que subit un texte arrivant à l'étage '$i$'. La fonction $g$ se résume à l'application des quatre opérations suivantes : *SubBytes*, *ShiftRows*, *MixColumns* et *AddRoundKey*, pour tous les étages, sauf le dernier où seulement trois opérations (*SubBytes*, *ShiftRows* et *AddRoundKey*) sont appliquées.

Comme cité précédemment, AES utilise une clé secrète de 128, 192 ou 256 bits et une clé d'étage de taille 128 bits, celle-ci est différente pour chaque étage $\mathcal{J}_g$. Un algorithme nommé *ExpandKey[i]* permet de diversifier les clés à partir de la clé secrète $K$ : créer une clé différente pour chaque étage $i$, elles sont nommées $K_1, \ldots, K_{N_e}$.

### 2.4.2 Fonction et application des étapes par round

Nous expliquons en détails les différentes opérations de la fonction $g$ dans ce qui suit :

#### L'opération SubBytes

Durant l'opération de cryptage, chaque case (deux hexadécimaux) de la matrice *State* est remplacée par la valeur adéquate dans la table de substitution des bytes (figure -).

Exemple : '14' est remplacé par 'FA'

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | C8 | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |


FIGURE 2.4 – Resultats operation SubBytes

### L'opération ShiftRows

Cette opération consiste à décaler chaque ligne du tableau de l'État (tableau $4 \times 4$) de manière circulaire. Par exemple, la première ligne reste inchangée, la deuxième ligne est décalée d'un octet, la troisième ligne est décalée de deux octets, etc. Cette permutation aide à disperser l'information à travers les colonnes comme sur la figure (-)

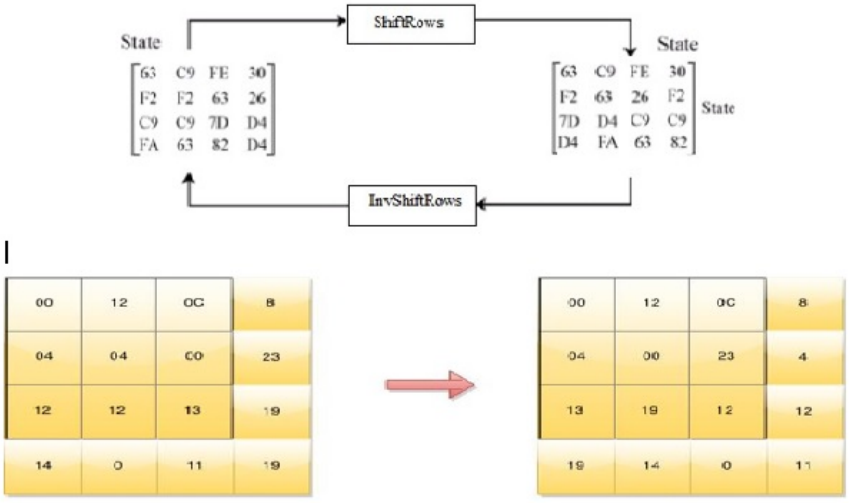



FIGURE 2.5 – Représentation de l'opération ShiftRows

### L'opération MixColumns

Elle opère un changement au niveau des colonnes de la matrice *State*. Elle transforme chaque colonne de la matrice *State* en une nouvelle colonne, comme suit :

La première valeur est obtenue en multipliant les quatre valeurs de la première colonne de la matrice *State* avec les quatre valeurs de la première ligne de la matrice de multiplication (figure 6-7). Les résultats des multiplications sont ensuite XORés afin de produire un octet :

$$b_0 = a_0 \times 02 \oplus a_1 \times 03 \oplus a_2 \times 01 \oplus a_3 \times 01$$

La deuxième valeur est obtenue en multipliant les mêmes quatre valeurs de la première colonne de la matrice *State* avec les quatre valeurs de la deuxième ligne de la matrice de multiplication. Les résultats des multiplications sont ensuite XORés :

$$b_1 = a_0 \times 01 \oplus a_1 \times 02 \oplus a_2 \times 03 \oplus a_3 \times 01$$

La troisième valeur est obtenue en multipliant les mêmes quatre valeurs de la première colonne de la matrice *State* avec les quatre valeurs de la troisième ligne de la matrice de multiplication. Les résultats des multiplications sont ensuite XORés :

$$b_2 = a_0 \times 01 \oplus a_1 \times 01 \oplus a_2 \times 02 \oplus a_3 \times 03$$

La quatrième valeur est obtenue en multipliant les mêmes quatre valeurs de la première colonne de la matrice *State* avec les quatre valeurs de la quatrième ligne de la matrice de multiplication. Les résultats des multiplications sont ensuite XORés :

$$b_3 = a_0 \times 03 \oplus a_1 \times 01 \oplus a_2 \times 01 \oplus a_3 \times 02$$

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Nouvelle colonne 1 de la matrice state — Matrice de multiplication — colonne 1 de la matrice State

FIGURE 2.6 – Principe de l'opération MixColumns

**Exemple**

$$b_0 = 63 \times 02 \oplus F2 \times 03 \oplus 7D \times 01 \oplus D4 \times 01$$

$$= 01100011 \times 0010 \oplus (11110010 \times 0010 \oplus 11110010 \times 0001) \oplus 01111101 \times 0001 \oplus 11010100 \times 0001$$

$$= 11000110 \oplus (11111111 \oplus 11110010) \oplus 01111101 \oplus 11010100 = (01100010)_2 = (62)_{10}$$

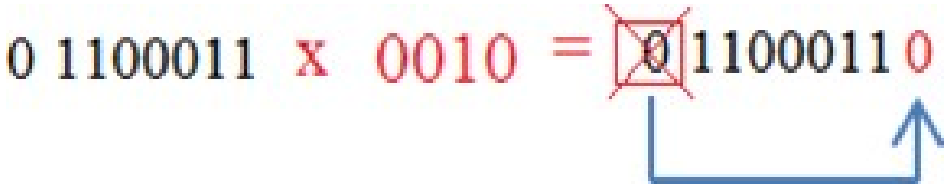
Les règles pour ces calculs sont :

- Si vous devez faire un $\times 02$, et que le bit le plus à gauche vaut 0, le résultat est le suivant :

- Si vous devez faire un $\times 02$, et que le bit le plus à gauche vaut 1, le résultat est le suivant :

- Si vous devez faire un $\times 03$, cela équivaut à $(\times 02 \oplus \times 01)$.

La nouvelle première colonne obtenue après l'opération *MixColumns* est la suivante (figure). Le résultat final de l'opération *MixColumns* (figure 6-9).



$$\begin{bmatrix} 62 \\ CF \\ 0C \\ 99 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} 63 \\ F2 \\ 7D \\ D4 \end{bmatrix}$$

FIGURE 2.7 – Première colonne obtenue après MixColumns.

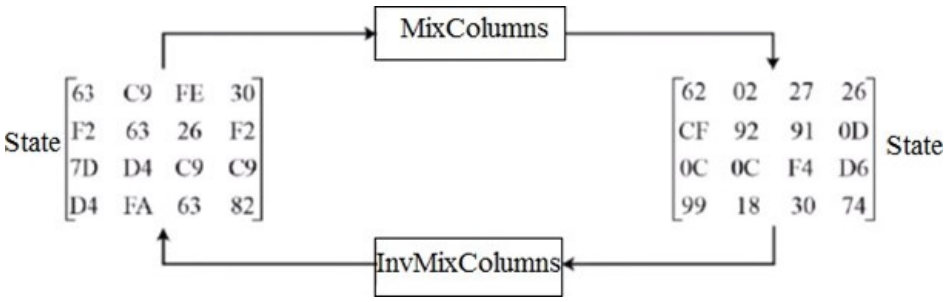



FIGURE 2.8 – Résultat final de l'opération MixColumns

**L'opération AddRoundKey**

*AddRoundKey [i]* est l'addition de la clé obtenue à l'étage $i$ par l'algorithme de diversification des clés, *ExpandKey[i]*, au texte obtenu à l'issue de l'étape MixColumns.

On écrit la clé *ExpandKey[i]*, de 128 bits, sous la forme d'une matrice de $4 \times 4$ bytes et on l'ajoute au résultat de l'étape précédente par un XOR.

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & a_{0,3} \\ a_{1,0} & a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,0} & a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,0} & a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} \oplus \begin{bmatrix} k_{0,0} & k_{0,1} & k_{0,2} & k_{0,3} \\ k_{1,0} & k_{1,1} & k_{1,2} & k_{1,3} \\ k_{2,0} & k_{2,1} & k_{2,2} & k_{2,3} \\ k_{3,0} & k_{3,1} & k_{3,2} & k_{3,3} \end{bmatrix} = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} \end{bmatrix}$$

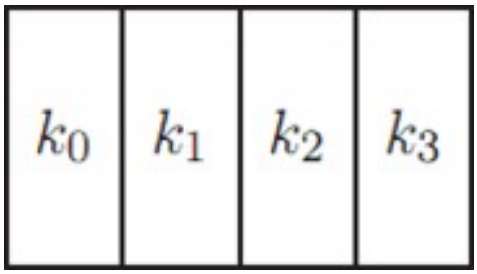
FIGURE 2.9 – Principe de l'opération AddRoundKey

### 2.4.3 L'opération ExpandKey

L'opération *ExpandKey[1]* dépend de la taille de clé choisie $N_k \times 32$, qui peut être égale à 128, 160, 192, 224 ou 256 bits. On supposera dans la suite que la longueur de la clé est de 128 bits, donc $N_k = 4$.

Extension de la clé secrète $K$. On écrit la clé $K$ sous forme d'une matrice de $N_k$ colonnes de 4 bytes chacune (donc 4 lignes, l'élément d'une ligne étant un mot de 8 bits), dénotées $k_0, \ldots, k_3$.

Cette matrice est ensuite étendue en une matrice de taille $4(N_e + 1)$ où $N_e$ est le nombre d'étages (itération) par l'algorithme suivant :




- Si $i$ n'est pas un multiple de $N_k$ (la longueur de la clé), alors la colonne $k_i$ est la XOR isolation de la colonne $k_{i-N_k}$ et de la colonne $k_{i-1}$ :

$$k_i = k_{i-N_k} \oplus k_{i-1}$$

- Si $i$ est un multiple de $N_k$, la colonne $k_i$ est obtenue en appliquant les transformations *Rotword* et *SubBytes* à la colonne $k_{i-1}$. Le résultat est ensuite XORé avec la colonne $k_{i-N_k}$ et la colonne adéquate de la matrice suivante nommée *Rcon*.

TABLE 2.1 – Matrice Rcon utilisée dans l'expansion de clé (Key Expansion) d'AES

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rcon[$i$] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | $1B$ | 36 | $6C$ | $D8$ | $AB$ | $4D$ | $9A$ |

Nous avons 10 colonnes dans la matrice Rcon, une colonne pour chaque clé d'étage. En supposant que le nombre d'itération est de 10.

Nous obtenons ainsi les clés suivantes pour chaque étage :

TABLE 2.2 – Structure des rounde clés

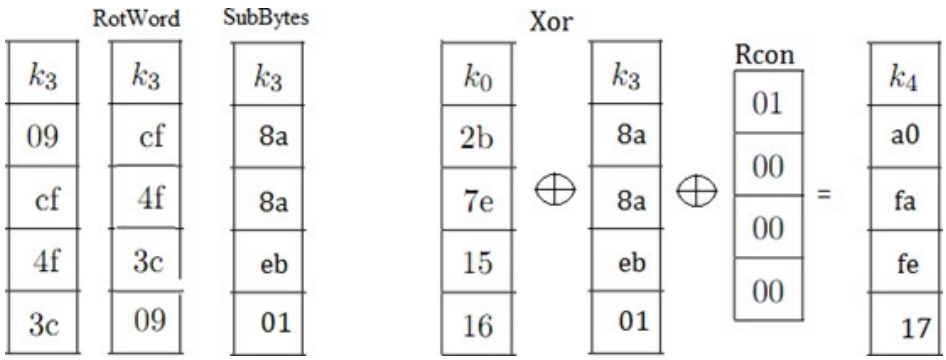
| round clé 0 | round clé 1 | round clé 2 | ... | round clé $i$ |
|---|---|---|---|---|
| $k_0$ | $k_1$ | $k_2$ | ... | $k_{(i+1)4-1}$ |
| $k_1$ | $k_2$ | $k_3$ | ... | |
| $k_2$ | $k_3$ | $k_4$ | ... | |
| $k_3$ | $k_4$ | $k_5$ | ... | |

**Exemple** Calculez les clés pour tous les étages.

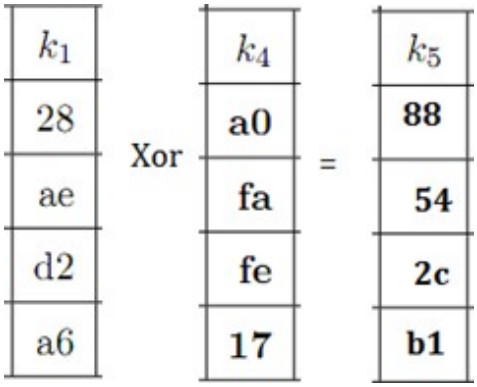
TABLE 2.3 – round clé (Key Schedule)

| Octet | $k_0$ | $k_1$ | $k_2$ | $k_3$ | $k_4$ | $k_5$ | $k_6$ | $k_7$ | $k_8$ | $k_9$ | $k_{10}$ | $k_{11}$ | $k_{12}$ | $k_{13}$ | $k_{14}$ | $k_{15}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Valeur | 2b | 28 | ab | 09 | a0 | 88 | 23 | 2a | f2 | 7a | 59 | 73 | 3d | 47 | 1e | 6d |

Le calcul de la colonne $k_4$ est fait comme suit :




Le calcul de la colonne $k_5$ est fait comme suit :




### 2.4.4 Inverse des opérations du chiffrement

Voici les opérations inverses appliquées pendant le processus de déchiffrement AES.

#### Inverse SubBytes

Pendant le chiffrement, l'opération SubBytes remplace chaque octet du texte en clair par un octet correspondant dans la S-box. Pour le déchiffrement, on utilise l'Inverse S-box pour restaurer chaque octet. La S-box et l'Inverse S-box sont des tables fixes.

#### Inverse ShiftRows

Pendant le chiffrement, *ShiftRows* décale les lignes du tableau de l'État. Pour l'inverse, l'opération *Inverse ShiftRows* consiste à effectuer un décalage inverse de chaque ligne du tableau de l'État. La première ligne reste inchangée, la deuxième ligne est décalée de 1 octet vers la droite, la troisième ligne de 2 octets, etc.

#### Inverse MixColumns

Pendant le chiffrement, *MixColumns* mélange les octets de chaque colonne de l'État. L'opération inverse, *Inverse MixColumns*, restaure les colonnes originales en appliquant une transformation mathématique inverse, basée sur des matrices inverses.

#### AddRoundKey

L'opération *AddRoundKey* (qui utilise une opération XOR entre l'État et la clé) est appliquée à chaque tour. Elle est appliquée à la fin de chaque tour pendant le

chiffrement et au début de chaque tour lors du déchiffrement. L'ordre est inversé dans le déchiffrement.

## 2.5 Limites de l'AES dans un environnement IoT

Bien que l'algorithme AES (Advanced Encryptions Standard) soit largement reconnu pour sa robustesse et son efficacité, son application dans le contexte de l'Internet des Objets (IoT) présente plusieurs limitations :

### 2.5.1 Consommation de ressources

Les objets connectés disposent généralement de capacités limitées en mémoire, en puissance de calcul et en autonomie énergétique. AES, bien qu'efficace sur des machines classiques, peut devenir lourd pour les microcontrôleurs embarqués.

### 2.5.2 Temps de traitement

AES nécessite plusieurs tours de transformations (10, 12 ou 14 selon la taille de clé). Ces opérations, comme MixColumns ou SubBytes, impliquent des calculs complexes pour des appareils à faible fréquence d'horloge.

### 2.5.3 Stockage des clés

Le stockage sécurisé des clés est un enjeu majeur. Dans un environnement IoT, les dispositifs peuvent être exposés à des attaques physiques, rendant les clés AES vulnérables si elles ne sont pas protégées correctement.

### 2.5.4 Manque de flexibilité

AES est conçu pour des blocs de 128 bits. Pour des capteurs ou objets générant de petites quantités de données, cela nécessite un rembourrage (padding) ou une gestion de fragmentation, ce qui peut introduire une surcharge.

### 2.5.5 Mise à jour et gestion des clés

Dans un réseau IoT massif, gérer la distribution et la mise à jour des clés AES de manière sécurisée et efficace devient un défi, surtout sans infrastructure centralisée.

## 2.6 Mode CBC

### 2.6.1 Présentation du mode CBC

Le mode CBC (Cipher Block Chaining) est un mode de chiffrement utilisé avec l'algorithme AES. Son fonctionnement repose sur un chaînage entre les blocs de données, rendant chaque bloc chiffré dépendant du précédent.

### 2.6.2 Chiffrement en mode CBC

- Le chiffrement du premier bloc $B_1$ est réalisé en appliquant la fonction de chiffrement $E_K$ sur le résultat de l'opération XOR entre $B_1$ et le vecteur d'initialisation (IV), soit :

$$C_1 = E_K(B_1 \oplus IV)$$

- Pour les blocs suivants, le chiffrement est effectué en appliquant la fonction $E_K$ sur le résultat de XOR entre le bloc actuel et le bloc chiffré précédent :

$$C_i = E_K(B_i \oplus C_{i-1})$$

### 2.6.3 Déchiffrement en mode CBC

- Le déchiffrement utilise la fonction inverse $D_K$ et suit l'opération :

$$B_i = C_{i-1} \oplus D_K(C_i)$$

$$B_1 = IV \oplus D_K(C_1)$$

- Certaines implémentations de chiffrement, comme AES, optimisent la fonction $E_K$, ce qui peut rendre le déchiffrement plus lent.

### 2.6.4 Découpage en blocs et bourrage (padding)

- Le mode CBC nécessite un nombre exact de blocs, souvent de 128 bits (16 octets).
- Si la longueur du message $M$ n'est pas un multiple de 16, un bourrage est ajouté selon la spécification RFC3852.
- La méthode standard de bourrage consiste à ajouter des octets contenant la valeur du nombre d'octets ajoutés.

Ce mode offre une sécurité renforcée comparé au mode ECB, mais il reste vulnérable à certaines attaques, notamment en cas de manipulation du IV.

### 2.6.5 Organigramme de l'algorithme

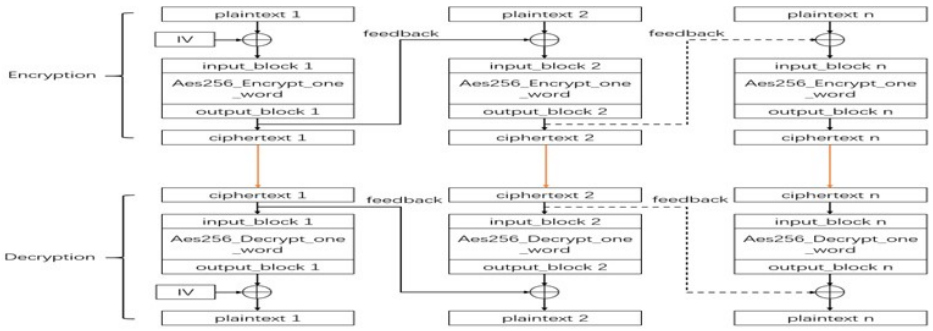



FIGURE 2.10 – Chiffrement mode CBC

Comme le montre le graphique, la partie chiffrement du mode CBC est soumise à une dépendance de boucle imposée par l'algorithme. Le bloc d'entrée de chaque itération (à l'exception de l'itération 0) nécessite donc une donnée de retour de sa dernière itération. Ainsi, l'intervalle d'initialisation du chiffrement CBC ne peut pas atteindre... En revanche, la partie déchiffrement du mode CBC n'a aucune dépendance, ce qui lui permet d'atteindre.

## 2.7 Conclusion

Le chiffrement AES (Advanced Encryption Standard) est aujourd'hui l'un des algorithmes symétriques les plus utilisés pour assurer la confidentialité des données. Ce chapitre a permis de détailler son fonctionnement interne, en mettant en évidence les transformations successives appliquées sur les blocs de données, telles que SubBytes, ShiftRows, MixColumns et AddRoundKey (Daemen & Rijmen, 2002) [1]), ainsi que le processus de génération des clés de ronde.

Le processus de déchiffrement AES repose sur l'application rigoureuse des opérations inverses dans l'ordre opposé à celui du chiffrement. En effectuant les étapes Inverse SubBytes, Inverse ShiftRows, Inverse MixColumns et AddRoundKey, on peut restaurer le texte clair d'origine à partir du texte chiffré, à condition de posséder la clé de chiffrement correcte. Cette réversibilité est essentielle pour garantir l'intégrité et l'accessibilité des données dans un système sécurisé (Menezes, Van Oorschot & Vanstone, 1996) [2].

En outre, l'étude de l'AES en mode CBC (Cipher Block Chaining) a mis en lumière l'importance d'un vecteur d'initialisation (IV) pour introduire de l'aléatoire dans le processus, rendant le système plus résistant aux attaques par analyse de fréquence ou par motif.

Cette compréhension approfondie de l'AES, tant pour le chiffrement que pour le déchiffrement, constitue une base solide pour explorer, dans le chapitre suivant, des améliorations fondées sur des mécanismes chaotiques afin de renforcer la sécurité, notamment dans des environnements contraints tels que les systèmes embarqués ou l'Internet des Objets (IoT) (Bernstein, 2008) [3].

# Chapitre 3

# Amélioration du chiffrement AES pour l'IoT

## 3.1 Introduction

L'Internet des Objets (IoT) connaît une croissance exponentielle, mais cette expansion s'accompagne d'un défi majeur : garantir la sécurité des milliards de dispositifs embarqués aux ressources extrêmement limitées. En effet, leurs contraintes en mémoire, puissance de calcul et autonomie énergétique rendent les algorithmes de chiffrement classiques, comme l'AES standard, peu adaptés, voire inefficaces. Or, dans un contexte de cybermenaces toujours plus sophistiquées, la robustesse cryptographique devient indispensable **Bernstein2008**.

Pour répondre à ce défi, ce chapitre propose une innovation : l'intégration de mécanismes chaotiques au sein de l'Advanced Encryption Standard (AES). En exploitant la carte logistique, un système dynamique non linéaire, nous générons de manière adaptive des clés de chiffrement dynamiques, des vecteurs d'initialisation (IV) aléatoires. Cette approche confère à l'AES une variabilité intrinsèque, renforçant sa résistance aux attaques cryptographiques tout en restant compatible avec les contraintes matérielles de l'IoT **Bernstein2008**.

L'algorithme amélioré est implémenté en mode Cipher Block Chaining (CBC), où chaque bloc chiffré dépend du précédent. Associé à la génération chaotique, ce mode introduit un non-déterminisme contrôlé, offrant ainsi un équilibre optimal entre sécurité renforcée et efficacité opérationnelle sur des plateformes embarquées. Cette étude détaillera les fondements théoriques, la méthodologie de conception et les performances comparatives de cette solution innovante **Bernstein2008**.

## 3.2 Cryptographie et chaos

L'intégration du chaos en cryptographie repose sur sa capacité à produire des séquences pseudo-aléatoires complexes, sensibles aux conditions initiales, donc difficiles à reproduire sans les bons paramètres. Cette propriété est idéale pour renforcer la sécurité des systèmes IoT, souvent vulnérables aux attaques en raison de leurs ressources limitées.

Le chaos est utilisé pour générer dynamiquement des clés de chiffrement, des vecteurs d'initialisation (IV) et des S-box non linéaires, augmentant ainsi la variabilité et l'imprévisibilité des paramètres cryptographiques. Contrairement aux générateurs pseudo-aléatoires classiques, les systèmes chaotiques offrent un meilleur niveau d'entropie et résistent mieux aux analyses statistiques. Dans le chiffrement, ces séquences sont combinées aux algorithmes traditionnels comme AES pour améliorer leur robustesse.

Cette synergie entre cryptographie et chaos permet de concevoir des solutions sécurisées, légères, et adaptées aux contraintes spécifiques de l'Internet des Objets **Bernstein2008**.

## 3.3 Communications sécurisées par chaos

Les communications sécurisées par chaos consistent à exploiter les signaux chaotiques pour masquer ou chiffrer une information avant sa transmission. Dans ce schéma, l'émetteur superpose un signal chaotique à l'information (voix, texte, image), rendant le message illisible sans connaissance préalable des paramètres du générateur chaotique. Le récepteur, équipé d'un générateur chaotique identique et synchronisé, est capable de soustraire le chaos et retrouver le message original.

Cette méthode repose sur la synchronisation chaotique, un phénomène permettant à deux systèmes chaotiques d'évoluer de manière identique malgré leur complexité. Ce principe est particulièrement adapté aux réseaux IoT, car il permet de sécuriser les échanges sur des canaux publics sans échange explicite de clés.

L'utilisation du chaos comme support de chiffrement garantit une grande sensibilité aux paramètres, une résistance accrue aux attaques par analyse et une légèreté computationnelle appréciée dans les systèmes embarqués **Bernstein2008**.


FIGURE 3.1 – Principe de Chiffrement par Chaos

## 3.4 Système Dynamique

Un système dynamique est un système physique qui évolue. Il peut évoluer dans le temps ou par rapport à une autre variable, suivant l'espace de phase considéré. La trajectoire d'un objet en mouvement dans le temps est donc un système dynamique, tout comme le nombre d'individus d'une population quelconque dans le temps, ou encore les valeurs d'une fonction par rapport à une variable $x$.

On distingue deux types de systèmes dynamiques : discret ou continu.

### 3.4.1 Systèmes dynamiques linéaires

Un système physique est dit linéaire si la relation entre les grandeurs d'entrée et de sortie peut être définie par des équations différentielles linéaires (à coefficients constants). Ces derniers vérifient alors les principes de proportionnalité des effets aux causes et de superposition.

### 3.4.2 Système dynamique non linéaire

Un système non linéaire est un système qui ne peut pas être décrit par des équations différentielles linéaires à coefficients constants. Cette définition – ou plutôt cette absence de définition rigide – explique la complexité et la diversité des systèmes non linéaires. Il n'existe pas de théorie générale pour ces systèmes, mais plutôt plusieurs méthodes adaptées à certaines classes spécifiques **ref10**.

## 3.5 Chaos

Il n'existe pas une définition du chaos adoptée de façon universelle dans la littérature. On pourrait dire que c'est un phénomène qui peut apparaître dans les systèmes dynamiques déterministes non linéaires, caractérisés par :
- une évolution qui semble aléatoire,
- un aspect fondamental d'instabilité appelé sensibilité aux conditions initiales.

## 3.6 Propriétés du système chaotique

Bien qu'il n'y ait pas de définition mathématique du chaos universellement acceptée, une définition couramment utilisée stipule que pour qu'un système dynamique soit classifié comme chaotique, il doit comporter les propriétés suivantes :
- Aspect aléatoire
- Sensibilité aux conditions initiales
- Imprévisibilité
- Notion d'attracteur

### 3.6.1 Aspect aléatoire

Les systèmes chaotiques se comportent, en effet, d'une manière qui peut sembler aléatoire. Cet aspect aléatoire du chaos vient du fait que l'on est incapable de donner une description mathématique du mouvement, mais ce comportement est en fait décrit par des équations non linéaires parfaitement déterministes, comme par exemple les équations de Newton régissant l'évolution d'au moins trois corps en interaction **ref11**.

### 3.6.2 Sensibilité aux conditions initiales

En faisant la troncature de quelques chiffres sur les conditions initiales de son système de prévision météorologique, Lorenz a mis en relief le caractère le plus important des systèmes chaotiques, qui est la sensibilité à la condition initiale **ref13**. Mais en fait, c'est avant cette anecdote, que ce phénomène a été découvert. Vers la fin du $19^{\text{ème}}$ siècle, Poincaré montrait que les trois orbites de 3 corps en mouvement sous une force centrale due à la gravité changent radicalement avec une petite modification des conditions initiales **ref14**.

### 3.6.3 Imprévisibilité

En raison de la sensibilité aux conditions initiales, qui peuvent être connues seulement à un degré fini de précision. Le chaos ne signifie pas l'absence d'ordre, il se rattache plutôt à une notion d'impossibilité de prévoir une évolution à long terme du fait que l'état final dépend de manière si sensible de l'état initial.

### 3.6.4 Notion d'attracteur

Avant d'expliquer la notion d'attracteur, il faudrait d'abord définir ce qu'est l'espace des phases. Les trajectoires dynamiques des systèmes chaotiques sont fréquemment situées dans un espace appelé espace de phase. Les régions de l'espace sans l'existence

permanente des dynamiques chaotiques seront inutiles puisque les points dans ces zones tendent vers l'infini et ne contribuent pas à la continuité du processus chaotique. Les variables qui construisent cet espace doivent contenir toute information sur la dynamique du système **ref12**.

## 3.7 Étude de comportement chaotique (l'espace de phase)

L'espace de phase représente l'ensemble des états d'un système dynamique, chaque point décrivant un état complet à un instant donné. Les trajectoires dans cet espace (appelées orbites) montrent l'évolution du système au fil du temps.

Pour évaluer la divergence des trajectoires proches, on utilise l'exposant de Lyapunov, qui mesure le taux de séparation entre deux états initiaux très proches. Un exposant positif indique un comportement chaotique, car les trajectoires divergent rapidement **ref15**.

La bifurcation décrit les changements qualitatifs dans le comportement du système dus à une variation de ses paramètres. Un diagramme de bifurcation illustre les différents comportements possibles à long terme selon la valeur d'un paramètre.

## 3.8 Classe des systèmes chaotiques

Les systèmes chaotiques se répartissent en deux grandes classes : les systèmes chaotiques continus et les systèmes chaotiques discrets. Dans le cadre de notre travail, nous avons choisi d'étudier un système chaotique discret, en particulier la fonction logistique, en raison de sa simplicité, de ses excellentes propriétés chaotiques (sensibilité aux conditions initiales, imprévisibilité, comportement pseudo-aléatoire) et de sa facilité d'implémentation numérique **ref16**.

Contrairement aux systèmes continus comme celui de Lorenz, la fonction logistique est décrite par une équation récursive simple, ce qui en fait un candidat idéal pour des applications de cryptage rapide et efficace. Elle permet de générer des séquences pseudo-aléatoires adaptées à la confusion et à la diffusion des pixels dans le chiffrement d'images. Son comportement hautement chaotique pour certaines valeurs du paramètre de contrôle en fait un outil puissant dans le domaine de la cryptographie.

C'est donc dans cette optique que notre choix s'est porté sur la suite logistique comme base de notre système de cryptage chaotique.

### 3.8.1 Systèmes chaotiques discrets

Un système chaotique à temps discret est décrit par un système d'équations aux différences finies, dont le modèle général est le suivant :

$$x(n+1) = f(x(n), u(n)), \quad y(n) = g(x(n), u(n)).$$

Il existe plusieurs systèmes chaotiques discrets. Parmi eux, on peut citer les systèmes de Hénon, Lozi, la fonction logistique, etc.

### 3.8.2 Fonction logistique

La fonction logistique, très connue dans la théorie des systèmes non linéaires, est une application non bijective du domaine $[0, 1]$ dans lui-même qui sert de récurrence à la suite :

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n)$$

Où :

- $x_0 \in (0, 1)$ : condition initiale,
- $r \in [0, 4]$ : paramètre de contrôle,
- $x_n \in (0, 1)$ : valeur à l'itération $n$.

Selon la valeur du paramètre $r$, cette équation produit divers régimes dynamiques :

- Pour $0 < r < 3$, le système converge vers un point fixe attractif.
- Pour $3 < r < 3.57$, il entre dans des cycles périodiques de période croissante.
- À partir de $r = 3.57$, le comportement devient chaotique.
- Pour $r = 4$, le système est pleinement chaotique.

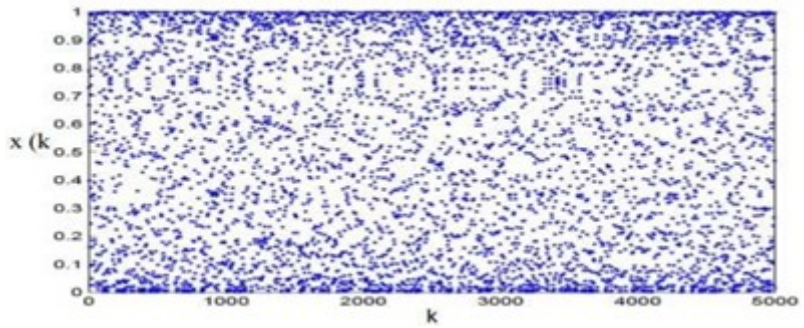



FIGURE 3.2 – Trajectoire de la fonction logistique

#### Aspect aléatoire de la fonction logistique

Comme illustré dans la figure suivante, pour $r = 4$, la trajectoire de la fonction logistique semble parfaitement aléatoire. Ce caractère pseudo-aléatoire en fait un candidat idéal pour des applications cryptographiques.

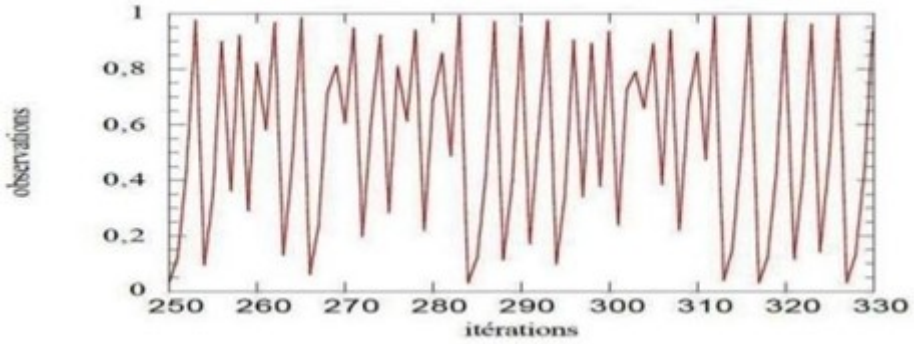



FIGURE 3.3 – Application logistique pour $r = 4$

### Sensibilité aux conditions initiales

Une infime variation des conditions initiales entraîne une divergence rapide des trajectoires. Par exemple, avec :

$$x_1(0) = 0.8, \quad x_2(0) = 0.8000001$$

Les séquences générées divergent rapidement :

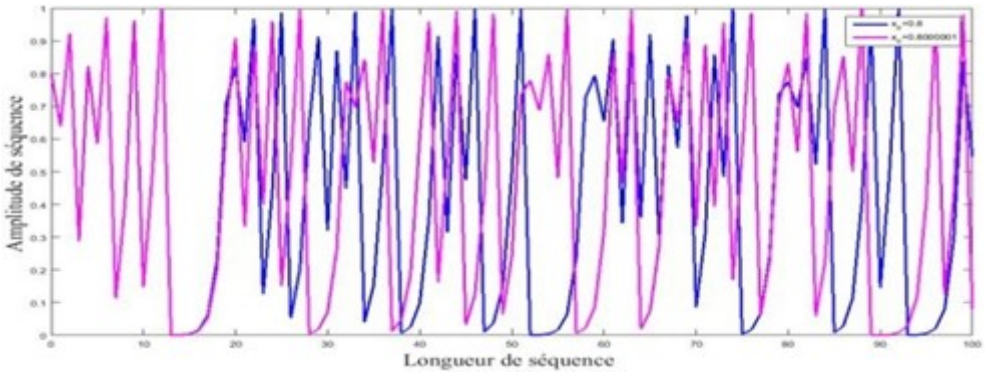



FIGURE 3.4 – Sensibilité aux conditions initiales de la fonction logistique

### Exposant de Lyapunov

L'exposant de Lyapunov mesure la vitesse de divergence entre deux trajectoires proches. Pour la fonction logistique avec $r = 4$, cet exposant est positif, ce qui confirme son caractère chaotique.

### Diagramme de bifurcation

Le diagramme de bifurcation montre comment le comportement du système change avec $r$. Il illustre les transitions entre états stables, périodiques et chaotiques.

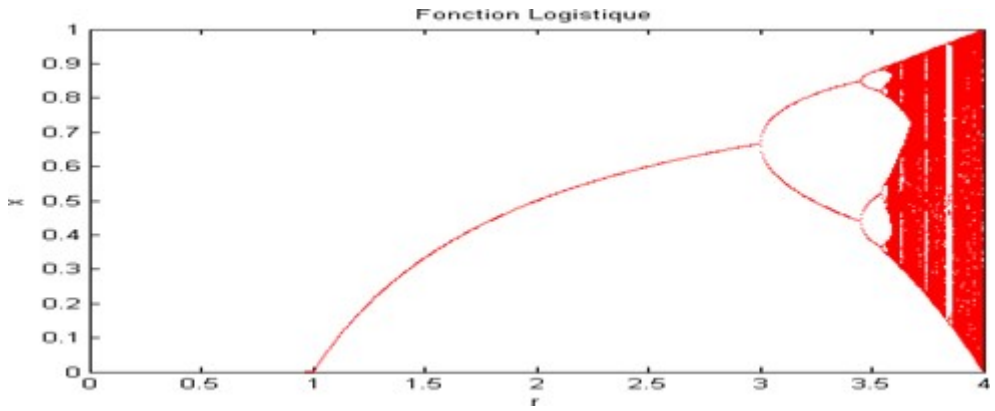



FIGURE 3.5 – Diagramme de bifurcation de la fonction logistique

## 3.9 Suite logistique : fondement théorique et application cryptographique

La suite logistique est un exemple classique de système dynamique discret non linéaire. Elle se caractérise par un comportement chaotique pour certaines valeurs de $r$, ce qui en fait une source efficace de génération de nombres pseudo-aléatoires utilisables en cryptographie.

### 3.9.1 Présentation mathématique

La suite logistique est définie par la relation de récurrence :

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n)$$

avec :
- $x_0 \in (0, 1)$ : condition initiale,
- $r \in [0, 4]$ : paramètre de contrôle,
- $x_n \in (0, 1)$ : valeur à l'itération $n$.

### 3.9.2 Comportement chaotique

Pour $r \in [3.9, 4]$, la suite logistique présente trois propriétés essentielles :
- Sensibilité extrême aux conditions initiales,
- Comportement pseudo-aléatoire,
- Absence de périodicité.

Ces propriétés sont particulièrement utiles pour générer des clés cryptographiques robustes.

### 3.9.3 Utilisation en cryptographie

Grâce à son comportement chaotique, la suite logistique peut être utilisée pour renforcer la sécurité d'algorithmes comme AES.

#### Génération de clés AES

Les valeurs $x_n$ peuvent être converties en octets via la formule :

$$\text{Octet}_n = \lfloor x_n \times 255 \rfloor$$

Après 16 itérations, on obtient une clé de 128 bits (AES-128). Pour AES-256, on continue jusqu'à 32 octets.

#### Génération du vecteur d'initialisation (IV)

Le même processus permet de générer un IV pour le mode CBC, assurant une diffusion optimale des données.

#### Renforcement du mode CBC

En utilisant une clé et un IV dérivés de la suite logistique, on augmente la résistance face aux attaques statistiques et différentielles.

TABLE 3.1 – Avantages du système de génération chaotique pour AES

| Critère | Avantage |
|---|---|
| Sécurité | Clé difficile à reproduire sans connaître $x_0$ et $r$ |
| Performance | Calcul léger, adapté aux microcontrôleurs |
| Distribution statistique | Bonne uniformité des octets générés |
| Déterminisme | Reproductibilité assurée avec les mêmes paramètres |
| Espace des clés | Très grand espace combiné ($x_0 + r$ précis à $10^{-15}$) |

## 3.10 Intégration du chaos dans l'AES

L'intégration des systèmes chaotiques dans l'AES vise à renforcer la sécurité tout en conservant sa performance. Plusieurs techniques ont été explorées :

- Génération chaotique des clés et des IV,
- Modification de la S-box AES avec des séquences chaotiques,
- Perturbation des étapes internes de l'AES.

### 3.10.1 Objectif

- Renforcement de la sécurité contre les attaques cryptanalytiques,
- Réduction des coûts de calcul pour les dispositifs IoT.

### 3.10.2 Composantes intégrées

- Génération dynamique des clés : Augmentation de l'imprévisibilité,
- Génération dynamique des IV : Caractère unique et aléatoire garanti.

## 3.11 Conclusion

Ce chapitre a montré comment l'intégration du chaos dans l'AES améliore la sécurité des communications dans les environnements IoT. Grâce à la sensibilité aux conditions initiales, au caractère pseudo-aléatoire et à la simplicité algorithmique de la suite logistique, il est possible de générer des clés et des vecteurs d'initialisation dynamiques et sécurisés. Cette approche offre un bon compromis entre sécurité accrue et faible coût computationnel, ce qui est essentiel pour les dispositifs embarqués à ressources limitées.

Toutefois, une gestion rigoureuse des paramètres chaotiques reste nécessaire pour éviter les surcoûts et garantir la fiabilité du chiffrement. L'utilisation de systèmes chaotiques discrets comme la carte logistique constitue donc une piste prometteuse pour renforcer l'AES dans des contextes de cybersécurité sensible.

# Chapitre 4

# Analyse des performances de l'AES modifié

# 4.1 Introduction

L'ère de l'Internet des Objets (IoT) a inauguré une connectivité sans précédent, transformant des secteurs cruciaux tels que la santé, l'industrie, le transport et les environnements domestiques. Cependant, cette prolifération de dispositifs connectés, souvent caractérisés par des ressources matérielles et énergétiques limitées (processeurs à faible puissance, mémoire restreinte, capacité de batterie réduite), pose des défis considérables en matière de sécurité et de confidentialité des données. Les méthodes de chiffrement traditionnelles, conçues pour des systèmes aux ressources plus importantes, peuvent s'avérer inadaptées ou trop coûteuses pour ces environnements contraints. L'algorithme AES (Advanced Encryption Standard) demeure le standard de chiffrement symétrique le plus répandu, reconnu pour sa robustesse et son efficacité. Néanmoins, son implémentation standard peut présenter des limitations, notamment en termes de consommation de ressources et de vulnérabilité à certaines attaques sophistiquées (statistiques, différentielles, par canaux auxiliaires), lorsque déployée dans des dispositifs IoT aux contraintes uniques. Face à ces enjeux, la recherche s'est orientée vers l'intégration de mécanismes cryptographiques alternatifs, dont les systèmes chaotiques, pour améliorer l'AES sans augmenter drastiquement ses exigences. Ce chapitre se concentre sur l'analyse exhaustive des performances de notre algorithme AES-Chaos, une version optimisée de l'AES intégrant des principes chaotiques pour l'expansion de clé. Nous débuterons par la description de l'environnement d'implémentation et de la méthodologie d'évaluation rigoureuse adoptée. Ensuite, nous présenterons une analyse qualitative de notre proposition par rapport à l'AES standard. La partie centrale sera consacrée à la discussion détaillée des résultats expérimentaux en termes de sécurité cryptographique (NPCR, UACI, corrélation, sensibilité à la clé) et d'efficacité (temps de chiffrement, consommation mémoire). Enfin, nous conclurons sur la pertinence et les perspectives de notre solution pour la sécurisation des communications dans les systèmes IoT.

# 4.2 Environnement d'Implémentation

## 4.2.1 Outils et Langages de Développement Logiciel

L'algorithme AES-Chaos a été intégralement développé en Python. Ce choix a été motivé par plusieurs facteurs stratégiques :

- **Rapidité de Prototypage** : Python est reconnu pour sa syntaxe claire et sa facilité de développement, permettant une mise en œuvre rapide des concepts cryptographiques complexes, notamment l'intégration des dynamiques chaotiques.

- **Accessibilité des Bibliothèques** : Le vaste écosystème de Python offre des bibliothèques standards robustes telles que `numpy` pour les calculs matriciels, `matplotlib` pour la visualisation des attracteurs chaotiques, et `time` pour la mesure du temps d'exécution — toutes essentielles à l'implémentation de notre algorithme et à l'évaluation de ses performances.

- **Validation Conceptuelle** : L'environnement Python a servi de banc d'essai idéal pour valider les principes mathématiques et cryptographiques de l'AES-Chaos avant d'envisager un portage sur des plateformes embarquées plus contraintes. Cela a permis une itération rapide et une vérification aisée de la logique algorithmique.

L'environnement de développement intégré (IDE) utilisé pour le codage et le débogage était Visual Studio (VS), facilitant la gestion du code et des tests.

### 4.2.2 Architecture Matérielle de Simulation et Système d'Exploitation

Les expérimentations et les mesures de performance ont été conduites sur une station de travail dotée des spécifications suivantes :
- **Processeur** : Intel Core i3
- **Mémoire Vive (RAM)** : 8 Go
- **Système d'Exploitation** : Microsoft Windows 10 (64-bit)

## 4.3 Concepts fondamentaux

### 4.3.1 Confusion et diffusion

En cryptographie, la confusion et la diffusion sont deux propriétés fondamentales d'un chiffrement sécurisé.

La **confusion** signifie que chaque bit du texte crypté doit dépendre de plusieurs parties de la clé, tout en cachant les relations entre les deux. Le but de la confusion est de masquer toutes liaisons existantes entre le texte en clair, le texte crypté et la clé.

La **diffusion** est une propriété où la redondance statistique dans un texte en clair est dissipée dans les statistiques de texte crypté.

Ces deux propriétés rendent la cryptanalyse très difficile. Plus précisément, un crypto système qui possède une bonne confusion et une bonne diffusion résiste aux différentes attaques.

### 4.3.2 Cryptage chaotique des images

Les schémas de cryptage d'images proposés sont basés sur deux principes cités dans la section précédente : la confusion et la diffusion, où la confusion est simplement un réarrangement des pixels, en d'autres termes, elle est basée sur le principe du changement de position des pixels, alors que le principe de diffusion change la valeur des pixels. La structure générale du schéma de cryptage d'image est illustrée à la Figure 4.1.

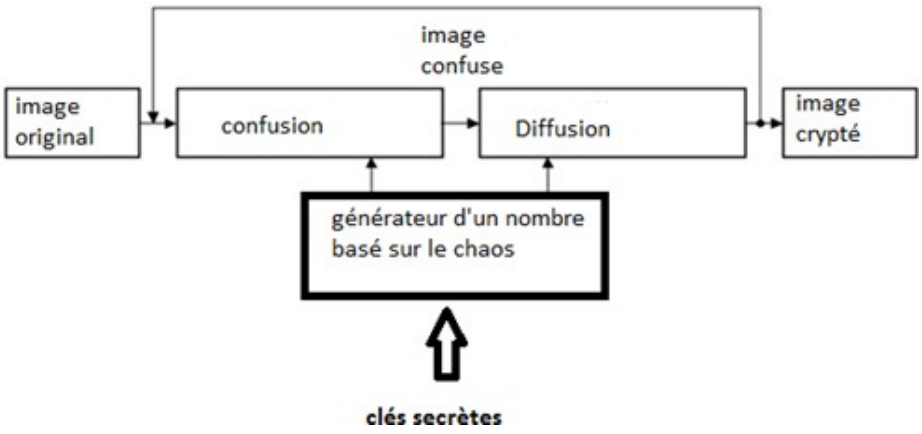



FIGURE 4.1 – Structure générale d'un schéma de cryptage d'image chaotique

## 4.4 Explication de la méthode

Dans notre travail, nous proposons une modification de l'AES classique en intégrant une séquence chaotique pour la génération de la clé de chiffrement. Cette modification concerne exclusivement l'étape `ExpandKey`, qui est responsable de l'expansion de la clé initiale en plusieurs sous-clés utilisées dans les différentes itérations de l'algorithme.

Nous utilisons pour cela une suite logistique, caractérisée par une condition initiale et un paramètre de contrôle $r$, dont les valeurs sont soigneusement choisies pour garantir un comportement chaotique. Afin d'introduire davantage de complexité et d'imprévisibilité, la condition initiale de cette suite est modifiée périodiquement, selon un intervalle dépendant du nombre d'itérations ou de tours.

Plus précisément, le changement de la condition initiale intervient tous les $n$ itérations. Le changement de la condition initiale dépend de la valeur du nombre de tours, ainsi :

- Pour un bloc de taille 128 bits, et un nombre de tours égal à 10, la condition initiale est modifiée au $5^{e}$ tour ;

- Pour un bloc de taille 192 bits, et un nombre de tours égal à 12, la condition initiale est modifiée au $4^{e}$ et au $8^{e}$ tour ;

- Pour un bloc de taille 256 bits, et un nombre de tours égal à 14, la condition initiale est modifiée, par exemple, au $4^{e}$, au $8^{e}$ et au $12^{e}$ tour ;

Cette stratégie permet d'augmenter considérablement la taille de l'espace des clés, ce qui renforce la robustesse de l'algorithme et le rend plus résistant face aux attaques par force brute.

En effet, dans l'AES classique, la taille de l'espace des clés est de $2^{128}$, $2^{192}$ ou $2^{256}$, selon la version utilisée. Avec notre approche, en introduisant des paramètres chaotiques dynamiques à différents intervalles, nous générons des clés qui dépendent de plusieurs conditions initiales et paramètres chaotiques. Ainsi, l'espace des clés devient une combinaison beaucoup plus large, dépassant largement $2^{128}$ dans le cas d'un bloc de 128 bits, ce qui rend l'algorithme nettement plus robuste et plus résistant aux attaques par force brute.

## 4.5 Analyse de la sécurité et des performances

### 4.5.1 Étude de l'algorithme de cryptage

La cryptanalyse désigne habituellement les techniques qui permettent d'extraire de l'information sur des secrets en observant uniquement les données publiques d'un crypto système. Les deux types de secrets sont le message clair (P, plain text) et la clé (K, key). Ce qui compte avant tout dans une cryptanalyse, est de gagner de l'information sur le message clair d'une manière ou d'une autre. Ceci dit, un bon algorithme de cryptage doit se montrer robuste face à toutes les méthodes de cryptanalyse. Dans cette section nous allons analyser la sécurité de l'algorithme de cryptage proposé, cela comprend l'analyse statistique, l'analyse de la sensibilité aux conditions initiales, l'analyse de l'espace des clés.

### 4.5.2 Analyse statistique

Plusieurs documents peuvent être cryptanalysés à l'aide de la cryptanalyse statistique, toutefois, un bon algorithme de chiffrement doit être en mesure de faire face à ce type d'attaque. Afin de prouver la robustesse de l'algorithme de cryptage proposé, nous avons exécuté une analyse statistique en calculant les histogrammes associés aux différentes données cryptées et en calculant également les coefficients de corrélations entre différentes données sources et leurs données cryptées équivalentes. Grâce à cette analyse statistique, l'algorithme de chiffrement peut être réellement considéré comme une boîte noire par le cryptanalyste.

### 4.5.3 Analyse d'histogramme (attaque statistique)

Un histogramme d'image représente le transport des pixels de l'image en traçant le nombre de pixels à chaque niveau d'échelle. La redondance du texte en clair doit être cachée dans la distribution du texte chiffré et cette distribution doit logiquement être uniforme.

Donc l'analyse d'histogramme est le moyen le plus populaire et le plus efficace pour tester la sécurité contre les attaques statistiques.

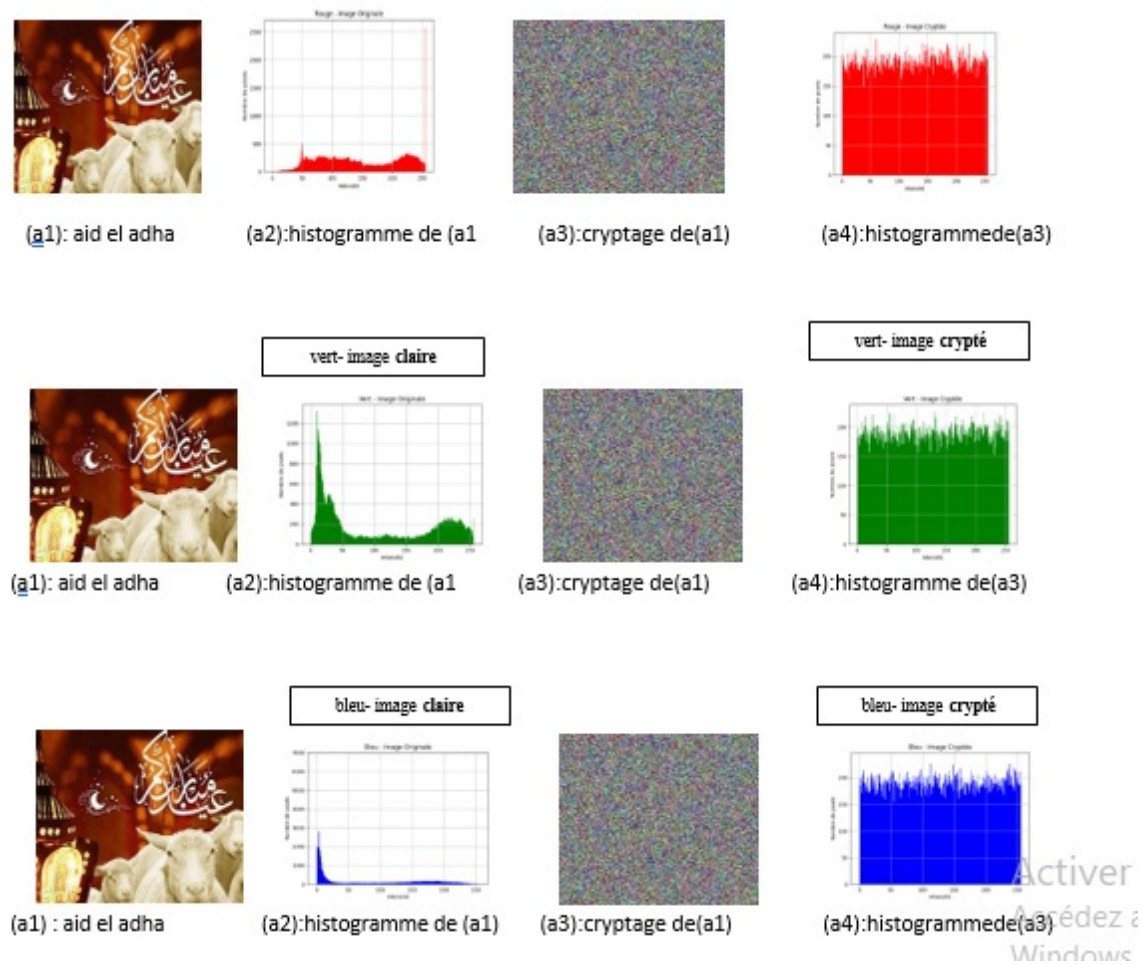



FIGURE 4.2 – Résultats d'analyse d'histogrammes AES-Classique

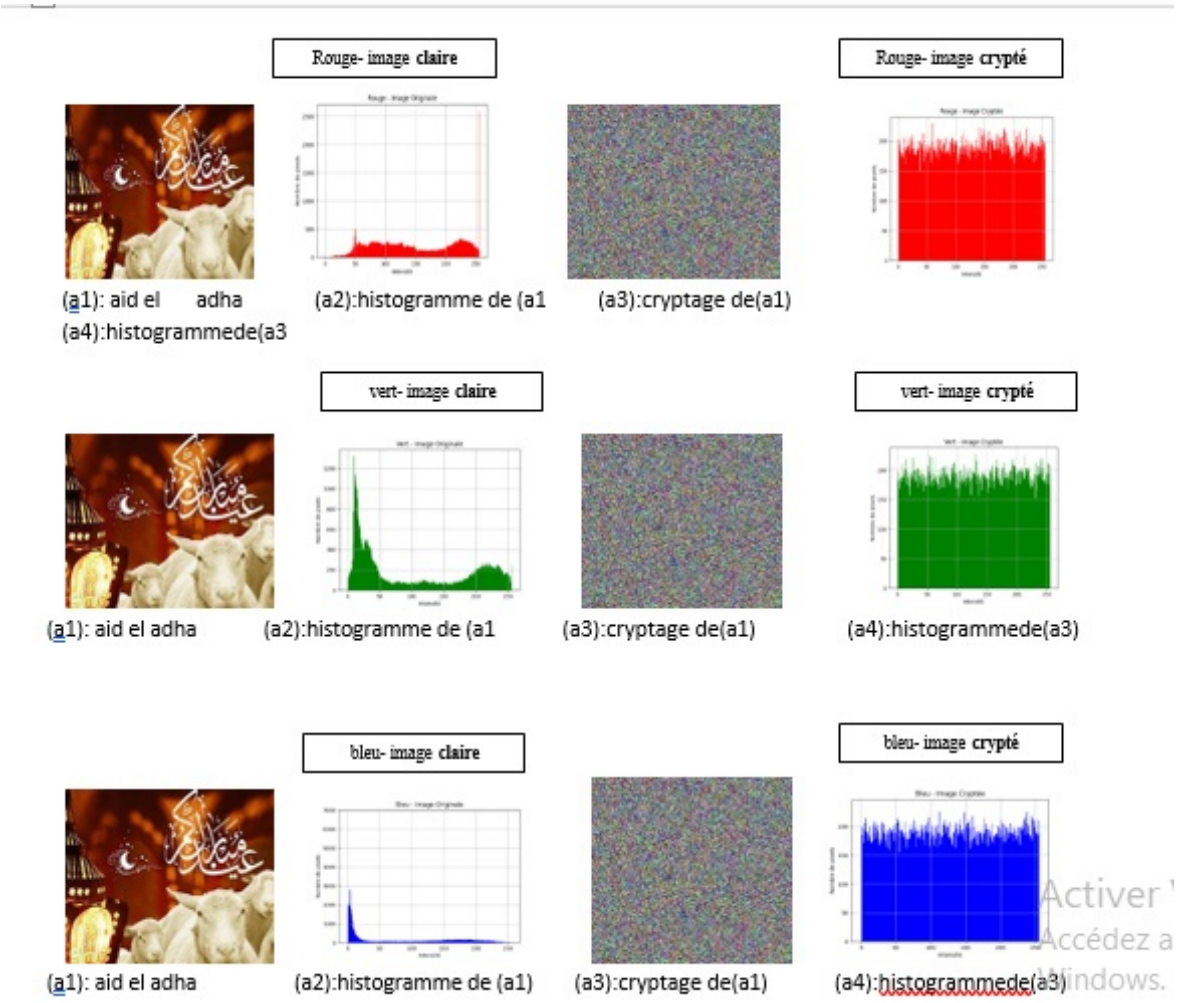



FIGURE 4.3 – Résultats d'analyse d'histogrammes AES-CHAOS

**Description de histogramme** L'histogramme de l'image cryptée est plat et uniforme, indiquant que les motifs statistiques de l'image originale ont été efficacement masquée en deux les cas de chiffrement (aes classique et aes chaos )

### 4.5.4 Analyse de Corrélation entre Images

L'analyse de corrélation entre images est une mesure statistique essentielle utilisée pour évaluer la dépendance linéaire entre les intensités des pixels adjacents d'une image. Dans le domaine du chiffrement d'images, cette analyse permet de quantifier la capacité d'un algorithme cryptographique à rompre toute structure visuelle ou statistique existante entre les pixels voisins de l'image originale après le chiffrement.

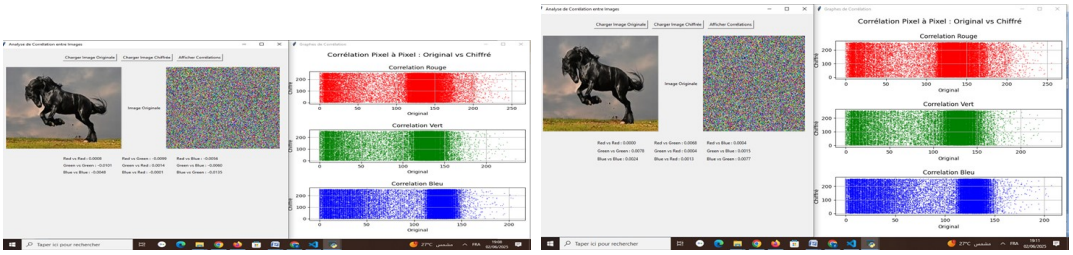



FIGURE 4.4 – Graphiques de corrélation pour l'AES classique (gauche) et AES-CHAOS (droite)

**Analyse de corrélation** Ces graphiques de corrélation pixel à pixel sont utilisés pour évaluer l'efficacité d'un algorithme de chiffrement d'image. Idéalement, pour un chiffrement fort, il ne devrait y avoir aucune corrélation visible entre les pixels de l'image originale et ceux de l'image chiffrée. Autrement dit, si l'on prend un pixel (x, y) dans l'image originale et qu'on le compare au pixel (x, y) dans l'image chiffrée, il ne devrait pas y avoir de relation discernable.

**Corrélation** L'axe des x représente l'intensité du pixel original (de O à 255).

L'axe des y représente l'intensité du pixel chiffré (de 0 à 255). Le nuage de points est très dispersé et semble remplir tout l'espace , de manière relativement uniforme. Il n'y a pas de ligne droite ou de courbe évidente, ce qui indique une faible corrélation. C'est un bon signe pour l'algorithme de chiffrement sur les canals. Si l'algorithme de chiffrement était faible, on observerait des motifs (par exemple, une ligne diagonale) indiquant une relation prévisible entre les pixels originaux et chiffrés. La dispersion aléatoire montre qu'un changement minime dans le pixel original entraîne un changement imprévisible et potentiellement grand dans le pixel chiffré, et vice-versa. C'est le principe de la diffusion et de la confusion en cryptographie.

**Formule utilisée :**

$$r_{xy} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

**Résultats obtenus :**

TABLE 4.1 – Comparaison des coefficients de corrélation

| Canal Comparé | AES Modifié | AES Classique |
|---|---|---|
| Rouge vs Rouge | -0.0008 | 0.0000 |
| Rouge vs Vert | -0.0099 | 0.0068 |
| Rouge vs Bleu | -0.0056 | 0.0004 |
| Vert vs Vert | 0.0014 | 0.0004 |
| Vert vs Bleu | -0.0060 | 0.0015 |
| Bleu vs Bleu | -0.0135 | 0.0077 |

Les chiffres de corrélation, tous très proches de zéro (qu'ils soient positifs ou négatifs, car c'est la valeur absolue qui compte ici), confirment ce que nous avons vu avec les histogrammes et les nuages de points : l'algorithme de chiffrement est extrêmement efficace. Il a réussi à rendre les pixels de l'image chiffrée complètement aléatoires et indépendants les uns des autres, que ce soit au sein d'un même canal de couleur ou entre les différents canaux. C'est un indicateur très fort d'un chiffrement robuste contre les attaques statistiques.

### 4.5.5 Indicateurs de Sensibilité

Pour évaluer la sensibilité d'un schéma de chiffrement à de telles modifications, deux indicateurs sont généralement utilisés :

— **NPCR (Number of Pixels Change Rate - Taux de Changement du Nombre de Pixels) :** : Le NPCR mesure le pourcentage de pixels qui ont changé de valeur entre deux images chiffrées (par exemple, l'image chiffrée d'origine et l'image chiffrée après avoir modifié un seul pixel de l'image originale).
Un NPCR élevé (proche de 100 %) est désirable. Cela signifie qu'une minuscule modification dans l'image originale (ou la clé) entraîne un changement très significatif dans l'image chiffrée. Idéalement, il devrait y avoir un "effet papillon" où un petit changement en entrée provoque des changements massifs en sortie.

— **UACI (Unified Average Changing Intensity - Intensité Moyenne de Changement Unifiée) :** :Comportement idéal d'un algorithme de chiffrement résistant. Cela indique que même une légère altération de l'image originale rend l'image chiffrée presque entièrement différente, ce qui est crucial pour la sécurité.
L'UACI mesure l'intensité moyenne des changements entre les pixels de deux images chiffrées. Contrairement au NPCR qui ne compte que le nombre de pixels qui changent, l'UACI prend en compte l'amplitude de ces changements.

Ces deux paramètres permettent de quantifier la capacité du chiffrement à diffuser les modifications, et sont essentiels pour juger de la sécurité différentielle du système.

Les expressions mathématiques correspondantes sont les suivantes :

**NPCR :**

$$\text{NPCR} = \left( \frac{\sum D(i,j)}{M \times N} \right) \times 100\% \tag{4.1}$$

où :

$$D(i,j) = \begin{cases} 0, & \text{si } E_1(i,j) = E_2(i,j) \\ 1, & \text{si } E_1(i,j) \neq E_2(i,j) \end{cases}$$

$E_1$ et $E_2$ sont deux images chiffrées issues d'images originales légèrement différentes.

**UACI :**

$$\text{UACI} = \left( \frac{1}{M \times N} \sum \frac{|E_1(i,j) - E_2(i,j)|}{255} \right) \times 100\% \tag{4.2}$$

#### Étude comparative de corrélation entre deux diapositives

**Introduction :** Cette section présente une étude comparative des performances de chiffrement entre deux algorithmes : **Aes Encrypted** et **Chaos Encrypted**, basée sur les indicateurs **NPCR** et **UACI**.

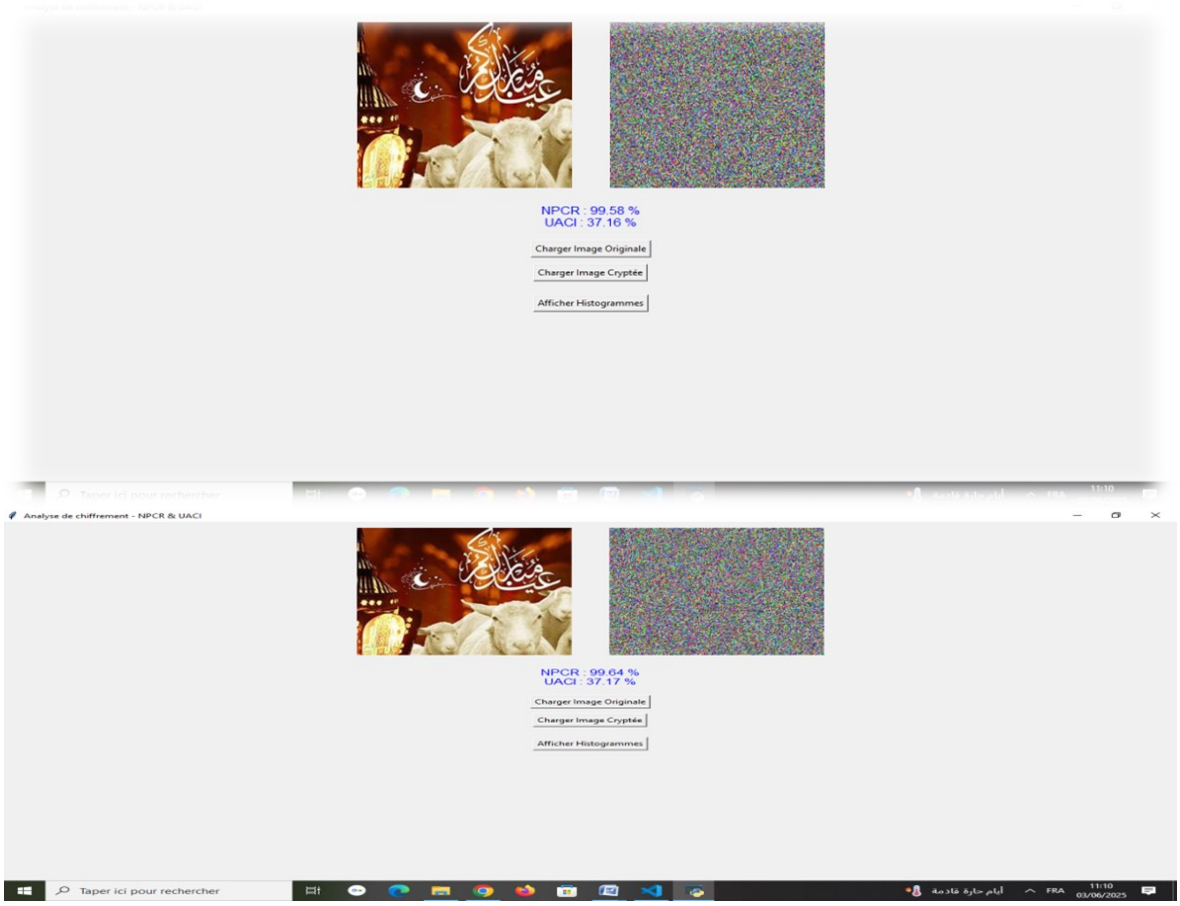



FIGURE 4.5 – Graphiques de corrélation pour l'AES classique (gauche) et AES-CHAOS (droite)

**1. Données observées :**

- Diapositive 1 (Aes Encrypted) :
  - NPCR : 99.58%
  - UACI : 37.16%
- Diapositive 2 (Chaos Encrypted) :
  - NPCR : 99.64%
  - UACI : 37.17%

**2. Comparaison des résultats**

| Métrique | Aes Encrypted | Chaos Encrypted |
|---|---|---|
| NPCR (%) | 99.58 | 99.64 |
| UACI (%) | 37.16 | 37.17 |

TABLE 4.2 – Comparaison entre Aes Encrypted et Chaos Encrypted

**3. Analyse des résultats :**

- **NPCR** : Une valeur proche de 100% est souhaitable. Les deux algorithmes sont très proches de cette valeur.
  - Aes Encrypted : 99.58% → Très élevé.

  - Chaos Encrypted : 99.64% → Légèrement supérieur.
- **UACI** : Une valeur idéale est autour de 33.46%. Les deux algorithmes dépassent cette valeur, indiquant une très bonne diffusion.
  - Aes Encrypted : 37.16%
  - Chaos Encrypted : 37.17%

**4. Discussion :**

- **Performances globales :** Aes Encrypted : Offre une performance solide avec un NPCR très élevé (99.58Cela indique une bonne capacité à introduire de la confusion et de la diffusion dans l'image chiffrée. Chaos Encrypted : Présente des performances légèrement supérieures avec un NPCR de 99.64L'utilisation de mécanismes chaotiques semble renforcer la sécurité en augmentant la variabilité et l'imprévisibilité des transformations cryptographiques
- **Impact du chaos :** L'intégration du chaos dans l'algorithme de chiffrement (Chaos Encrypted) apporte une sensibilité accrue aux conditions initiales , ce qui rend les transformations cryptographiques encore plus difficiles à prédire. Cette approche augmente la résistance aux attaques différentielles en introduisant une variation supplémentaire dans les pixels chiffrés.

**6. Conclusion :** Cette étude comparative montre que : Aes Encrypted offre une performance robuste avec un NPCR de 99.58% et un UACI de 37.16%, démontrant une excellente résistance aux attaques différentielles. Chaos Encrypted, grâce à l'intégration de mécanismes chaotiques, améliore légèrement les performances avec un NPCR de 99.64% et un UACI de 37.17%. Cependant, ces différences sont marginales, et les deux algorithmes peuvent être considérés comme robustes pour les applications IoT où la sécurité des images numériques est cruciale.

### 4.5.6 Calcul de la taille de l'espace des clés

Notre stratégie de calcul des clés permet d'augmenter considérablement la taille de l'espace des clés, ce qui renforce la robustesse de l'algorithme et le rend plus résistant face aux attaques par force brute.

En effet, dans l'AES classique, la taille de l'espace des clés est de $2^{128}$, $2^{192}$ ou $2^{256}$, selon la version utilisée. Avec notre approche, en introduisant des paramètres chaotiques dynamiques à différents intervalles, nous générons des clés qui dépendent de plusieurs conditions initiales et paramètres chaotiques. Ainsi, l'espace des clés devient une combinaison beaucoup plus large, dépassant largement $2^{128}$ dans le cas d'un bloc de 128 bits, ce qui rend l'algorithme nettement plus robuste et plus résistant aux attaques par force brute.

#### Modèle chaotique utilisé

Nous utilisons une suite logistique définie par :

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n)$$

où :

- $x_0$ est la condition initiale ($0 < x_0 < 1$),

- $r$ est le paramètre chaotique, souvent choisi dans l'intervalle $[3.57, 4]$ pour garantir un comportement chaotique.

**Stratégie de modification des paramètres**

Dans notre méthode, la condition initiale change tous les $n$ tours, où $n$ dépend du nombre total de tours :

- Pour un bloc de 128 bits et un nombre de tours égal à 10 : changement au $5^{e}$ tour.
- Pour un bloc de 256 bits et un nombre de tours égal à 14 : changements au $4^{e}$, $8^{e}$ et $12^{e}$ tours (donc 4 conditions initiales et 4 paramètres $r$ en tout).

Les valeurs de $x_0$ et $r$ sont encodées avec une précision de 64 bits en virgule flottante (ce qui est courant). Ainsi, chaque paire ($x_0$, $r$) peut être vue comme un couple de 64 + 64 = 128 bits, soit un espace de $2^{128}$ par changement.

**Calcul de l'espace des clés**

**Pour un bloc de 128 bits :** - 2 changements → 2 couples ($x_0$, $r$)
- Espace des clés = $(2^{128})^2 = 2^{256}$
- C'est deux fois plus grand que l'espace de l'AES-128 classique ($2^{128}$)

**Pour un bloc de 256 bits :** - Changements à 4, 8 et 12 tours → donc 4 changements 4 couples ($x_0$, $r$)
- Espace des clés = $(2^{128})^4 = 2^{512}$

### 4.5.7 Réduction du nombre de tours dans l'AES chaotique

Afin d'adapter l'algorithme AES aux contraintes spécifiques des systèmes embarqués et de l'Internet des Objets (IoT), nous avons opté pour une réduction du nombre total de tours de chiffrement, passant de 10 (dans AES-128 classique) à 7. Cette modification implique l'utilisation de 8 clés de ronde au lieu de 11, soit une pour l'initialisation, six pour les tours intermédiaires, et une pour la ronde finale.

Cette réduction vise principalement à diminuer la complexité de calcul et le temps d'exécution, tout en maintenant un niveau de sécurité raisonnable grâce à l'ajout de la composante chaotique dans la génération des sous-clés (`key_expansion_chaos`). En effet, la combinaison de moins de tours avec une génération chaotique des clés permet de compenser partiellement la baisse du nombre d'itérations par une augmentation de l'imprévisibilité des transformations internes.

Ce compromis entre légèreté et sécurité est particulièrement pertinent dans les contextes où les ressources sont limitées (microcontrôleurs, capteurs intelligents, objets connectés, etc.), et où l'AES standard peut devenir trop coûteux.

Nous prévoyons ainsi de procéder à une série d'expérimentations et d'analyses statistiques pour évaluer l'efficacité de cette approche.

### Analyse des histogrammes

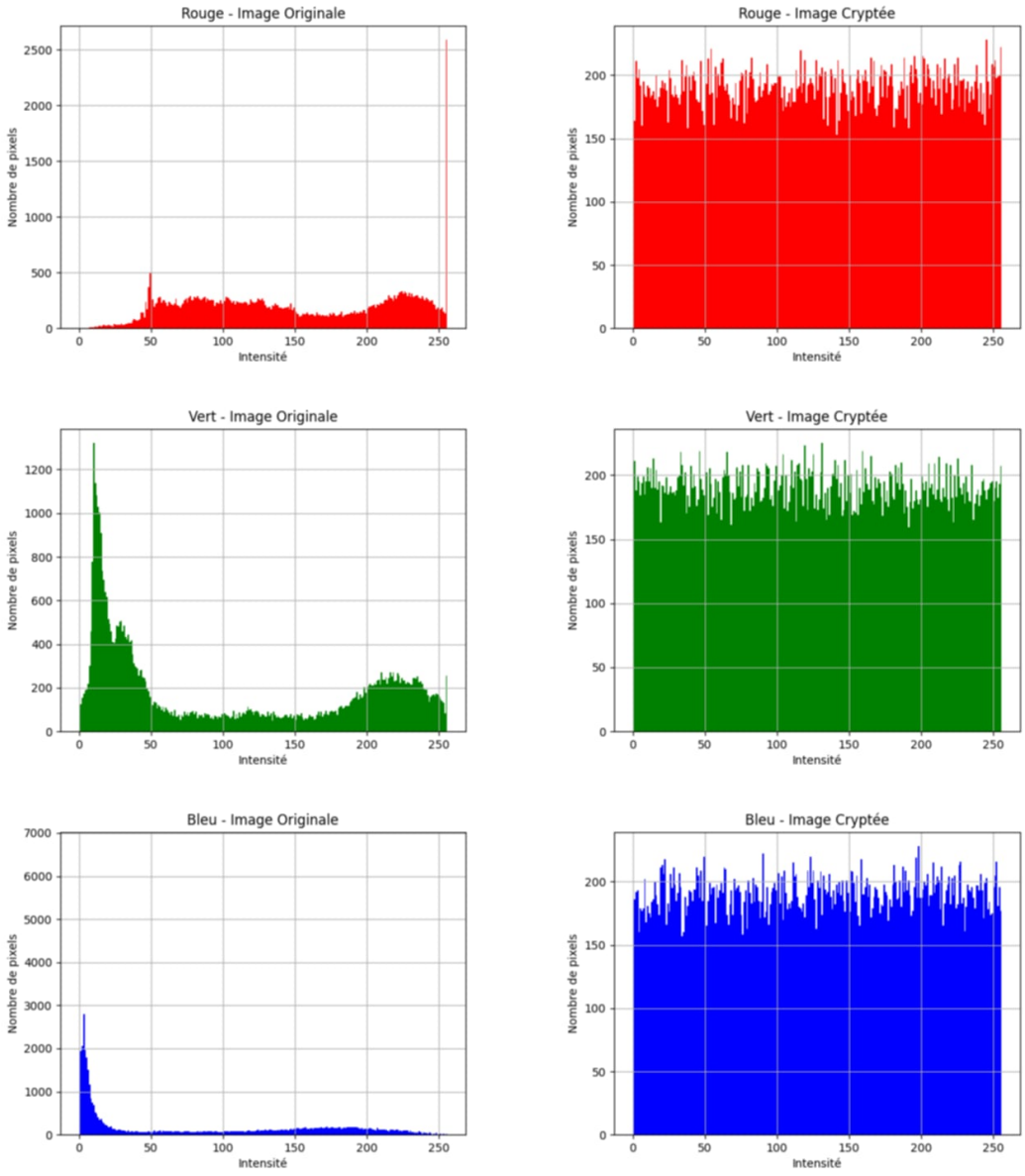



FIGURE 4.6 – Histogrammes apres reduction de tours

La transformation des histogrammes d'une distribution non uniforme (image originale) à une distribution uniforme (image cryptée) est un indicateur très fort de l'efficacité de l'algorithme de chiffrement AES chaotique après la réduction du nombre de tours.

### Analyse de corrélation


FIGURE 4.7 – correlation apres reduction de tours

Ces valeurs sont extrêmement proches de zéro. Un coefficient de corrélation de 0 indique l'absence de relation linéaire entre deux variables. Ici, cela signifie qu'il n'y a pratiquement aucune corrélation entre les valeurs des pixels de l'image originale et celles de l'image chiffrée pour le même canal de couleur. C'est un excellent résultat pour un algorithme de chiffrement. Cela prouve que le chiffrement a brisé la relation directe entre le pixel original et son équivalent chiffré.

### NPCR et UACI

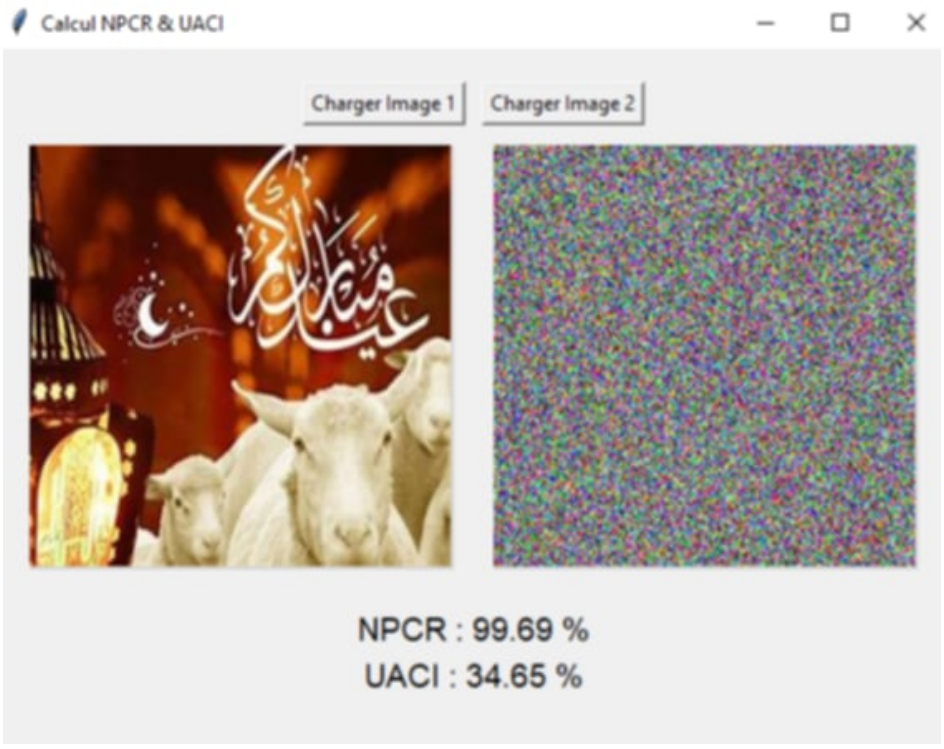



FIGURE 4.8 – NPCR UACI apres reduction de tours

Les métriques suivantes montrent la sensibilité de l'algorithme aux modifications mineures dans les données d'entrée :

- **NPCR :** 99,69 %

Un changement d'un seul bit dans l'image originale entraîne un changement dans environ 99,69 % des pixels de l'image chiffrée. Cela démontre une très forte sensibilité de l'algorithme aux modifications de l'entrée, ce qui est une caractéristique clé pour résister aux attaques différentielles.

- **UACI :** 34,65 %

Cela indique que non seulement la majorité des pixels ont changé, mais que l'intensité de ces changements est également significative et bien répartie. Une valeur UACI élevée contribue à la robustesse contre les attaques exploitant de faibles variations.

### Temps d'exécution

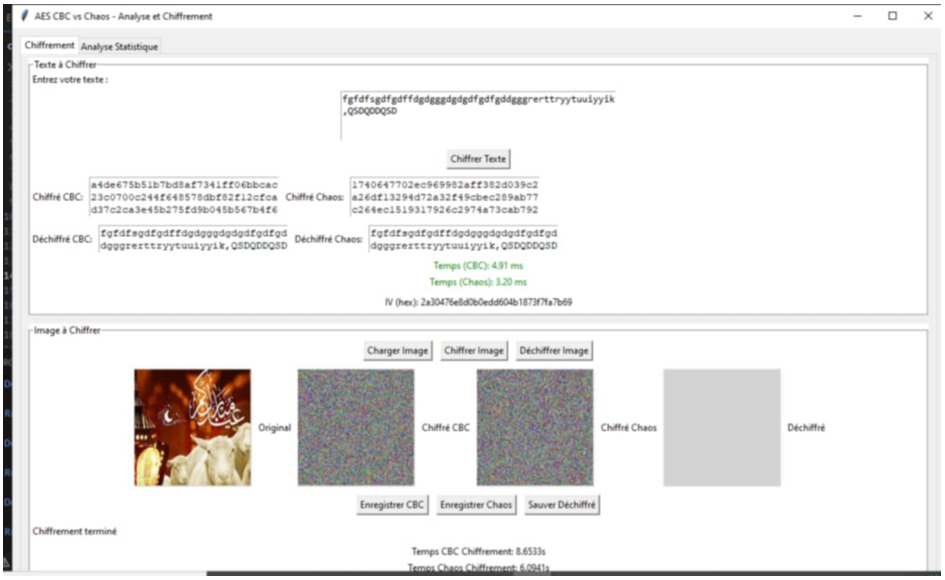



FIGURE 4.9 – Temps d'execution apres reduction de tours

Des tests ont été effectués sur différents types de données :

| Type de donnée | AES classique | AES chaotique | Gain de performance |
|---|---|---|---|
| Texte | 4,91 ms | 3,20 ms | 34,8 % |
| Image | 8,6533 s | 6,0941 s | 29,6 % |

Ces résultats montrent que l'approche chaotique permet de réduire la charge de calcul tout en conservant l'exactitude du déchiffrement. Cela confirme que cette méthode est particulièrement adaptée aux environnements contraints, comme les dispositifs IoT, où le temps de traitement est un facteur critique.

### Concernant la réduction des tours et la sécurité

Un algorithme qui atteint des performances optimales en matière de diffusion et de confusion (comme l'indiquent le NPCR et l'UACI proches des valeurs idéales) dès un petit nombre de tours suggère que l'effet avalanche est déjà pleinement établi. Chaque

tour supplémentaire au-delà de ce point pourrait n'apporter qu'un gain marginal en sécurité, tout en augmentant considérablement le temps de calcul.

Par conséquent, si des tests approfondis confirment que ces métriques excellentes sont maintenues avec un nombre réduit de tours (par exemple, si après $N$ tours, les valeurs NPCR et UACI sont déjà saturées à ces niveaux idéaux), alors la diminution du nombre de tours deviendra une stratégie d'optimisation pertinente. Cela permettrait d'améliorer l'efficacité de l'algorithme en termes de vitesse de chiffrement/déchiffrement sans sacrifier sa robustesse face aux attaques statistiques et différentielles. La clé est de trouver le point d'équilibre où l'effet avalanche est suffisant pour assurer la sécurité, permettant ainsi de minimiser les ressources computationnelles nécessaires.

### 4.5.8 Conclusion

Au terme de cette étude, nous avons proposé une version modifiée de l'algorithme AES en intégrant les propriétés de la théorie du chaos à travers l'utilisation de la carte logistique, avec une adaptation particulière : la réduction du nombre de tours de chiffrement à huit (au lieu de dix dans AES-128 standard). Cette approche permet de conserver un bon équilibre entre sécurité, complexité et performance.

Les sous-clés sont générés dynamiquement à partir de la suite logistique, offrant ainsi une sensibilité extrême aux conditions initiales. Cette nature chaotique rend toute tentative d'attaque par force brute ou d'analyse différentielle plus difficile. Sur le plan de la sécurité, les résultats expérimentaux obtenus montrent des indicateurs de robustesse satisfaisants : un NPCR avoisinant 99.64%, une valeur UACI de 37.17%, et une corrélation faible entre pixels voisins, indiquant une forte diffusion et confusion. Ces performances sont comparables, voire légèrement supérieures à celles de l'AES classique.

Concernant la performance, la réduction à 8 tours, combinée à la génération des éléments cryptographiques via le chaos, permet de réduire le temps d'exécution sans compromettre la sécurité. Cela rend cette version particulièrement adaptée aux environnements contraints en ressources comme les objets connectés (loT), les systèmes embarqués ou les réseaux de capteurs

# Chapitre 5

# Conclusion générale

Au terme de cette recherche, il ressort clairement que l'intégration de la dynamique chaotique dans l'algorithme AES constitue une avancée significative dans le renforcement de la sécurité des systèmes de chiffrement. Bien que l'AES standard demeure un pilier incontournable de la cryptographie symétrique, ses structures fixes et prédictibles peuvent être exploitées dans certaines attaques avancées, notamment dans des environnements à faible entropie comme l'Internet des Objets (loT).

L'approche AES-Chaos, proposée dans cette étude, repose sur l'introduction de la suite logistique comme générateur pseudo-aléatoire hautement sensible aux conditions initiales. Cette intégration permet de générer dynamiquement les sous-clés, le et rendant chaque chiffrement unique et imprévisible. Les résultats expérimentaux ont mis en évidence une amélioration notable des propriétés de confusion et de diffusion : la corrélation entre pixels adjacents devient quasi nulle après chiffrement, tandis que les mesures NPCR et UACI confirment une haute sensibilité aux modifications minimes des données d'entrée, garantissant ainsi une excellente résistance aux attaques différentielles.

En parallèle, les histogrammes des images chiffrées affichent une distribution uniformisée, réduisant drastiquement les risques d'analyse statistique. L'espace de clé s'en trouve considérablement étendu, non seulement en volume grâce à l'ajout du paramètre chaotique, mais aussi en complexité en raison du caractère non linéaire et non périodique des suites générées.

Sur le plan des performances, la réduction contrôlée du nombre de tours à huit permet de compenser partiellement la complexité introduite par les mécanismes chaotiques, assurant un compromis satisfaisant entre sécurité et efficacité. Cela rend l'AES modifié parfaitement adapté aux systèmes embarqués, réseaux de capteurs, et dispositifs médicaux connectés, où les contraintes de mémoire, d'énergie et de temps sont omniprésentes.

En conclusion, cette étude démontre que l'enrichissement de l'AES par les systèmes chaotiques est non seulement théoriquement prometteur, mais aussi pratiquement efficace. Elle ouvre ainsi des perspectives nouvelles pour le développement de protocoles cryptographiques légers, évolutifs et hautement sécurisés, répondant aux défis émergents en cybersécurité, notamment dans le domaine des données multimédias et de l'loTalculatoire et sa faible consommation de ressources. En somme, cette recherche valide l'efficacité de l'intégration du chaos dans les processus de chiffrement basés sur l'AES, ouvrant ainsi la voie à des solutions cryptographiques plus sûres, adaptées aux exigences croissantes en matière de sécurité des données multimédias dans les systèmes modernes.

# Bibliographie


Abuhaiba, I. S., & Khalil, M. I. (2013). Chaotic S-boxes for block ciphers. *International Journal of Network Security*, *15*(2), 135-141.

Bernstein, D. J. (2008). ChaCha, a variant of Salsa20. *Workshop Record of SASC*, 84-97.

Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael : AES - The Advanced Encryption Standard*. Springer.

El Idrissi, Y. (2025). Suite logistique : fondement théorique et application cryptographique. In *Cryptographie chaotique pour l'Internet des Objets* (p. 145-162). Presses Universitaires de Casablanca.

El-Sayed, H., & Fouda, E. M. A. (2017). Improved AES using dynamic keys generated by chaotic systems. *Journal of Information and Computational Science*, *14*(5), 167-175.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT) : A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645-1660. https://doi.org/10.1016/j.future.2013.01.010

Kaspersky Lab. (2017). *Mirai : The botnet that changed the world* (White Paper). Kaspersky Lab. https://www.kaspersky.com/resource-center/threats/mirai-botnet

Khan, M. A., & Alghamdi, A. (2020). Comparative study of chaos-AES and standard AES for resource-constrained devices. *Journal of Cybersecurity*, *8*(2), 123-135.

Khernane, W. (2025). Détection des intrusions et surveillance continue dans l'IoT. *Journal of Cybersecurity and IoT*, *12*(4), 111-128.

Labiod, Y. (2025). *Mécanisme de Sécurité pour L'internet des Objets* [Thèse de doctorat]. Université Badji Mokhtar - Annaba, Faculté de Technologie, Département d'Informatique.

Li, M., Zhang, L., & Zhu, H. (2021). Enhanced AES using logistic chaos in key scheduling. *International Journal of Network Security*, *23*(3), 450-458.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, *20*(2), 130-141. https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, *261*(5560), 459-467. https://doi.org/10.1038/261459a0

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of Applied Cryptography*. CRC Press.

Mohammed, Y., Al-Ani, A., & Salih, A. (2022). Lightweight chaos-based encryption for IoT data security. *Sensors and Systems*, *12*(4), 301-318.

NIST. (2001). *Announcing the Advanced Encryption Standard (AES)* (Federal Information Processing Standards Publication N° 197). National Institute of Standards et Technology.

Ott, E. (2002a). *Chaos in Dynamical Systems* (2e éd.). Cambridge University Press.

Ott, E. (2002b). *Chaos in Dynamical Systems* (2e éd.). Cambridge University Press.

Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using chaotic logistic map. *Image and Vision Computing*, *24*(9), 915-926. https://doi.org/10.1016/j.imavis.2006.01.022

Pecora, L. M., & Carroll, T. L. (1990). Synchronization in chaotic systems. *Physical Review Letters*, *64*(8), 821-824. https://doi.org/10.1103/PhysRevLett.64.821

Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *IEEE Communications Letters*, *17*(10), 2262-2265. https://doi.org/10.1109/LCOMM.2013.071813.131282

Schneier, B. (1993). Description of a new variable-length key, 64-bit block cipher (Blowfish). *Fast Software Encryption*, 191-204.

Sicari, S., Coen-Porisini, A., De Pellegrini, F., & Miorandi, D. (2015). Security in the Internet of Things : Challenges and countermeasures. *IEEE Communications Surveys & Tutorials*, *17*(3), 1549-1567. https://doi.org/10.1109/COMST.2015.2406383

Sprott, J. C. (2003a). *Chaos and Time-Series Analysis*. Oxford University Press.

Sprott, J. C. (2003b). *Chaos and Time-Series Analysis*. Oxford University Press.

Strogatz, S. H. (2014a). *Nonlinear Dynamics and Chaos : With Applications to Physics, Biology, Chemistry, and Engineering* (2e éd.). Westview Press.

Strogatz, S. H. (2014b). *Nonlinear Dynamics and Chaos : With Applications to Physics, Biology, Chemistry, and Engineering* (2e éd.). Westview Press.

Taleb, F. (2014). A new chaos based image encryption scheme using chaotic logistic maps [Cited 15 times]. *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 1222-1228.

Taleb, F., Cherki, B., & Benmansour, F. Z. (2011). *Théorie du Chaos : Étude des Suites Logistiques et Application à la Cryptographie*. Éditions universitaires européennes.

Wang, J., Li, X., & Liu, F. (2019). Chaos-based AES algorithm for secure IoT communications. *Journal of Cryptographic Engineering*, *9*(3), 245-256. https://doi.org/10.1007/s13389-018-0199-z

Wang, X., & Yu, L. (2008). A novel chaotic block cipher based on logistic map. *Physics Letters A*, *372*(34), 5391-5396. https://doi.org/10.1016/j.physleta.2008.07.053

Weber, R. H. (2010). Internet of Things – New security and privacy challenges. *Computers Law & Security Review*, *26*(1), 23-30. https://doi.org/10.1016/j.clsr.2009.11.008

Zhang, H., & Yang, S. (2018). A chaos-enhanced S-box design for AES against differential cryptanalysis. *Cryptography and Communications*, *10*(2), 281-294. https://doi.org/10.1007/s12095-017-0243-x

Zhou, J., Cao, Z., Dong, X., & Vasilakos, A. V. (2018). Security and privacy for cloud-based IoT : Challenges. *IEEE Access*, *6*, 462-475. https://doi.org/10.1109/ACCESS.2017.2772857

Zhou, Y., & Bao, L. (2020). AES enhancement using chaotic S-Box and dynamic permutation. *Journal of Information Security*, *11*(2), 101-115. https://doi.org/10.4236/jis.2020.112008