الجمـهورية الجزائرية الديمقراطية الشعبيـة
وزارة التعليم العالي والبحث العلمي
جـامعـة سعـيـدة د. مـولاي الطـاهـر
كـلـيـة التكنولوجيا
قسم: الإعلام الآلي

Mémoire de Master

Spécialité : Sécurité informatique et cryptographie

Thème

Cryptographie Légère pour l'internet des objets

Présenté par :
MOUADIH Abdelghani
LACHI Youcef

Dirigé par :
Dr Mekkaoui Kheireddine

Promotion 2021 - 2022

Remerciement

Tout d'abord, NOUS remercions **ALLAH** le tout puissant qui nous a donné la foi, qui nous a guidés durant toute notre vie et qui nous a donné la volonté de continuer nos études.

Nous exprimons aussi notre gratitude, la plus profonde à Monsieur **K.MEKKAOUI** qui a bien voulu nous confier ce sujet, et qui a assuré l'encadrement de ce travail, nous lui reconnaissons l'aide inestimable et ses conseils sans lesquels ce travail n'aurait pas abouti.

Nous souhaitons également remercier l'ensemble de nos enseignants durant les 5 années qui nous ont conduites à l'obtention de notre Master ; leurs entières disponibilités, Leurs aides et leurs conseils ont été pour nous un point fort dans la réussite de nos études ! !

MERCI…

Dédicace

MOUADIH Abdelghani

Rien n'est aussi beau à offrir le fruit d'un labeur qu'on dédie du fond du cœur à ceux qu'on aime et qu'on remercie en expriment la gratitude et la reconnaissance durant toute notre existence. Je dédie ce mémoire : Á ma chère maman qui à souhaiter vivre pour longtemps juste pour nous voir qu'est-ce que nous allons devenir, qui peut être fière et trouver ici le résultat de longues années de sacrifices et de Privations pour m'aider à avancer dans la vie, que dieu la protège. Á la bougie qui a éclairé mon chemin depuis ma naissance, à celle dont j'ai prononcé le premier mot, source de ma vie et de mon bonheur, que dieu la protège. À cet homme qui a consacré toute sa vie pour faire de moi un homme digne de ma place, dédicace à mon cher père. Á mes frères : qui n'ont cessé d'être pour moi des exemples de persévérance, de courage et de générosité, que dieu les garde pour moi. Á mes amis et amies par le monde qui n'ont cessé de m'encourager, qui m'ont assisté dans ces moments difficiles et m'ont servi d'exemple. Enfin, tous qui ont participé de près ou de loin pour l'accomplissement de mon travail.

Dédicace

LACHI Youcef

À celle qui m’a accordé la vie, attention et beaucoup d'affection, à ma douce maman. À cet homme qui a consacré toute sa vie pour faire de moi un homme digne de ma place, dédicace à mon cher père. A mon cher frère qui a veillé sur mon bonheur

À mes chers amis qui n'ont pas hésité de me remonter le moral et de m'avoir accompagné toutes ces années.

ملخص

هذا العمل هو تعميق الاتجاه الحالي "إنترنت الأشياء". لقد اتخذنا عناصر معينة مثل الأمان ولكننا طورنا أيضًا مفاهيم أخرى وخاصة تلك الخاصة بالتشفير الخفيف من أجل إجراء توليفة لاحقة على إنترنت الأشياء.

إن إنترنت الأشياء هو امتداد للإنترنت الحالي لجميع الكائنات التي يمكنها الاتصال بشكل مباشر أو غير مباشر بالأجهزة الإلكترونية المتصلة نفسها بالإنترنت. هذا البعد الجديد للإنترنت مصحوب بقضايا تكنولوجية واقتصادية واجتماعية وحوكمة كبرى.

الكلمات الدالة:

قمامة ؛ ماء البحر؛ التلوث البحري؛ معادن ثقيلة؛ منفذ الأثر البيولوجي

Abstract

This work is a deepening on the trend of the moment "Internet of Things". We have taken certain elements such as security but we have also developed other concepts and especially that of light cryptography in order to make a consequent synthesis on the IoT.

The Internet of Things is an extension of the current Internet to all objects that can communicate directly or indirectly with electronic equipment that is itself connected to the Internet. This new dimension of the Internet is accompanied by major technological, economic, societal and governance issues.

Key words:

IoT, Internet of Things, Light Cryptography, Arduino, Simulation.

Résumé

Ce travail est un approfondissement sur la tendance du moment ‘’Internet des objets’. Nous avons pris certains éléments comme la sécurité mais nous avons également développé d’autres concepts et surtout celui de la cryptographie légère afin de faire une synthèse conséquente sur l’IdO.

L’Internet des objets est une extension de l’Internet actuel à tous les objets pouvant communiquer de manière directe ou indirecte avec des équipements électroniques eux-mêmes connectés à l’Internet. Cette nouvelle dimension de l’Internet s’accompagne de forts enjeux en matière technologique, économique, sociétale et de gouvernance.

.

Mots clés:

IDO, Internet des objets, la cryptographie légère, Arduino, Simulation.

TABLE DES MATIERES

LISTE DES FIGURES ET TABLEAUX

CHAPITRE I

CHAPITRE II

CHAPITRE III

CHAPITRE IV

Introduction Générale

INTRODUCTION GENERALE

L'Internet Of Things désigne l'ensemble des réseaux constitués d'objets physiques qui sont incorporés à des logiciels, capteurs ou encore plusieurs technologies afin d'échanger des données avec d'autres dispositifs et systèmes sur Internet. Ça veut dire l'omniprésence de différents objets qui, à travers des schémas d'adressage, deviennent capables de réaliser des interactions les uns avec les autres et de coopérer avec leurs voisins pour atteindre des objectifs. Aujourd'hui, avec les évolutions des technologies et des réseaux, le champ des possibilités s'agrandit et l'IoT est de plus en plus présent dans divers aspects de notre quotidien.

On vise donc à rendre ces objets conscients ou ils se trouvent, quand et quoi faire. La montée en puissance de l'IOT devient de plus en plus visible et réelle dans plusieurs domaines dans la vie quotidienne. Tout ça résulte de grandes bénéfices, on obtient une meilleure gestion d'énergie, on affiche une amélioration dans le coté sanitaire, et même une simplification des tâches quotidiennes.

Pourtant, l'IOT ne vit que ses premiers temps, beaucoup de travail reste encore à faire surtout du côté des standardisations, de routage et d'identification, et surtout de sécurité. Néanmoins, puisque l'internet des objets vie quotidiennement avec nous, ça impose l'obligation de mettre en place des systèmes de sécurité forts et qui s'adaptent avec l'hétérogénéité des objets et leurs capacités limitées. En effet, on aura systématiquement l'obligation d'affronter les menaces classiques d'attaque qui touchent les données et les réseaux, mais aussi, de nouvelles menaces toucheront l''intégrité des objets et ses infrastructures que la vie privée des personnes.

Ce mémoire est organisé en quatre chapitres. Le premier chapitre sera consacré à la présentation de l'internet des objets, ainsi que l'introduction de quelques notions fondamentales utilisées dans le domaine de l'IOT.
Le deuxième chapitre, nous l'avons consacré à la sécurité et la cryptographie. Nous commencerons d'abord par la définition de la sécurité informatique ses différents aspects tels que la cryptologie, on s'intéressera bien évidemment à la cryptographie légère utilisée pour ce genre d'objets avec des ressources limitées et l'explication du principe qu'on a utilisé pour le déroulement de notre simulation. Ensuite, nous présenterons le Simulateur Arduino. Dans l'ultime chapitre nous allons réaliser une comparaison due aux résultats de notre implémentation.

Chapitre I :

Internet des objets

INTRODUCTION

L'Internet des objets ou IOT (Internet of Things), est un paradigme émergeant dans le monde des réseaux informatiques. Il peut être défini comme une évolution et extension de l'Internet de nos jours pour l'inclusion de tous les objets et les endroits dans notre entourage (réfrigérateurs, thermostat, maisons, véhicules, routes, etc.). Le concept prometteur de l'IoT va nous simplifier la vie, nous faire gagner du temps, décharger notre cerveau de la mémorisation de données logistiques (itinéraires, temps de prise des médicaments, etc.). Ainsi, l'accès ubiquitaire à différents types d'informations permettrait la sophistication du mode de vie et une amélioration significative de la qualité des services dans différents domaines.

L'IoT qui est une nouvelle vague de l'Internet, est en réalité une partie naissante de l'Internet du futur, appelé l'Internet de tous les objets ou IoE (Internet of Everything), qui vise à interconnecter les gens, les données et tous les objets, de telle sorte qu'il y ait une fusion entre le monde réel (physique) et le monde numérique (virtuel) ; les objets du monde physique vont être incorporés dans le monde virtuel de l'Internet. Cela fait appel à de nouvelles tendances et innovations que ce soit sur le plan architectures de communications ou sur le plan présentation et exploitation des services.

Ce chapitre est consacré à la présentation du domaine de l'Internet des objets et les aspects qui s'y rapportent [1].

I.1.Historique de l'internet des objets :

L'émergence de l'Internet des objets ce n'est qu'un résultat de convergence entre multiples technologies, à savoir l'Internet, la communication sans fil, les systèmes embarqués, systèmes microélectroniques et la nanotechnologie. Dans cette section, nous citons les évènements les plus marquants sur le chemin de la concrétisation de l'IoT. Le concept d'un réseau de dispositifs intelligents a été évoqué pour la première fois en 1982, avec le premier appareil connecté à Internet à l'Université Carnegie Melon capable de signaler à son inventaire si les boissons nouvellement chargées sont bien froides. Ainsi, en 1991, Mark Weiser a introduit l'informatique omniprésente à travers son papier intitulé : «L'ordinateur du 21ème siècle » et a présenté d'avance la vision contemporaine de l'Internet des objets. Un peu plus tard, en 1994, Steve Mann avait créé le WearCam qui était parmi les premières caméras à apparaitre sur le web. WearCam comporte les parties suivantes : (1) un groupe de caméras (ou uniquement une) qui sont fixées au corps, d'une manière quelconque, à deux mains libres (2) des moyens d'enregistrement, de traitement et de transmission des images capturées par les caméras (3) un moyen d'affichage qui a la capacité de présenter une image ou un flux d'images de l'appareil photo. Les images capturées seront communiquées vers une entité (une station de base) à la disposition de l'utilisateur. Ensuite, en 1998, l'informatique ubiquitaire a commencé d'attiré l'attention par le fait qu'elle permettrait l'incorporation flexible et efficace de l'informatique dans la vie quotidienne. Mark Weiser disait : « là où la réalité Chapitre 3 : généralités sur l'Internet des objets 47 virtuelle met l'humain en dedans du monde des ordinateurs, l'informatique ubiquitaire force plutôt l'ordinateur à s'instaurer dans le monde réel». En 1999, la désignation Internet des objets a été prononcée pour la toute première fois par Kevin Ashton. Après, en 2000 la société LG annonce son premier réfrigérateur intelligent connecté à Internet. De plus, la technologie RFID (Radio Frequency IDentification) qui est l'une des technologies constitutionnelles de l'IoT, a

commencé à être massivement déployée vers les années 2003 et 2004. D'autre part, une initiative très intéressante a été prise en 2008 ; un groupe de recherche appelé IPSo Alliance s'est consacré à promouvoir l'utilisation du protocole IP (Internet Protocol) pour les réseaux d'objets miniatures intelligents. De nombreux travaux de recherches ont été succédés et se sont tous concentrés autour de la réalisation, dans les meilleures conditions, de la vision de l'Internet des objets et la mener à sa maturité en dépit de tous les défis soulevés. Cela avec la considération des progrès technologiques continus dans le marché des dispositifs intelligents et dans le domaine de technologies de télécommunication (comme : le cloud computing, le concept du SDN (Software-Defined Networking), etc.) [2].

I.2. Typologie des objets

Avec l'avènement de l'Internet des objets, la connexion Internet acquiert une troisième dimension; en plus de la possibilité de se connecter n'importe quand et n'importe où, il est désormais possible d'être connecté avec n'importe quel objet. De plus, les objets connectés sont identifiés de façon unique et sont capable de récolter des informations environnementales (liées aux changements des paramètres de l'environnement, comme la température), comportementales (issues des variations d'état de l'objet lui-même ou des objets contextuels), de les traiter et de les communiquer sur Internet. D'où vient leur appellation par objets intelligents. CisCo prévoit que d'ici quelques années, spécifiquement en 2020,sera une réalité et le nombre d'objets connectés dépassera les 50 milliards [3]. A ce stade, il est nécessaire de noter que les données massives générées par un nombre immense d'objets intelligents connectés présente, partiellement, une source de la charge globale de données qualifiées de BigData sur Internet [4]. On distingue différents types de dispositifs connectés à l'IoT, ou qui font connecter d'autres objets à Internet, dont on cite principalement

I.2.1. Les objets d'identification

Codes barre, marqueurs RFID et autres dispositifs miniaturisés qui servent à l'identification et la traçabilité des objets sur lesquels ils sont collé pouvant être collés sur les objets d'usage courant (ex. vêtements, marchandises, livres, véhicules, etc.). Ce type de dispositifs nécessite qu'il y ait un lecteur pour récupérer leurs données qui seront par la suite téléchargées sur un serveur et deviennent alors accessibles via le système d'information d'une organisation ou directement sur Internet [3].

I.2.2. Les capteurs

Les capteurs dans l'IoT permettent de récolter des informations contextuelles concernant les objets dans lesquels ils sont intégrés, ou les environnements sur lesquels ils sont déployés. Les capteurs communiquent les informations collectées sur Internet d'une manière directe ou indirecte, tout dépend du modèle adopté pour l'intégration des réseaux de capteurs à l'internet.

I.2.3. Les sources d'énergie

Un drone désigne un aéronef miniature sans pilote, pouvant porter des charges utiles, communiquer et exécuter des commandes en toute flexibilité. Les drones sont utilisés dans des applications civiles aussi bien que dans des applications militaires pour accomplir des missions bien déterminées. On entend parler de l'efficacité de l'utilisation des drones dans le domaine commerciale pour par exemple, les livraisons à domicile des commandes faites sur Internet. Aussi, des opérations de sauvetage, d'exploration et de surveillance sont réalisables par les drones dans le contexte des applications militaires. Bien que la technologie (ou bien son prototype) des drones en elle-même existait depuis bien longtemps, son exploitation idéale dans différentes applications demeure modeste. Récemment, les drones sont élus pour faire une importante part de l'Internet du futur, soit en tant que objets intelligents terminaux rapportant des données de contrôle, soit en

tant que routeurs particuliers (mobiles et volants) de données entre les parties connectées à Internet. Comparés aux capteurs qui sont le plus souvent stationnaires ou dans certains cas mobiles mais dans tous les cas, manquent de l'aspect aérien, un drone parvient très efficacement à donner une vision aérienne sur l'état de la zone à contrôler même dans les zones isolée et/ou inaccessibles (là où il est difficile d'installer une infrastructure terrestre avec des points d'accès et des stations de base).

I.2.4. Smartphones et tablettes électroniques

Les smartphones et les tablettes qui sont déjà connectés à Internet par le biais de diverses technologies (Wi-Fi, 3G, 4G) permettent aux utilisateurs de communiquer à distances avec les autres types d'objets connectés dans l'IoT. Les objets intelligents peuvent rapporter en temps réel l'état actuel aux utilisateurs via Internet. Dans ce cas, les utilisateurs reçoivent des e-mails ou simplement des messages d'alertes sur leurs Smartphones ou tablettes, tout dépend de l'application. Il est meme possible que les utilisateurs supervisent ou ordonnent leurs objets connectés, à distance, via leurs smartphones ou tablettes [3].

La figure ci-dessous présente les principaux types d'objets dans l'IoT

Figure I.1. Typologie des objets dans l'IoT.

I.3. Cycle de vie d'un objet connecté dans l'IoT

Dans l'IoT, les objets intelligents passent par trois étapes : la phase préparatoire (bootstrapping, la phase opérationnelle et la phase de maintenance.

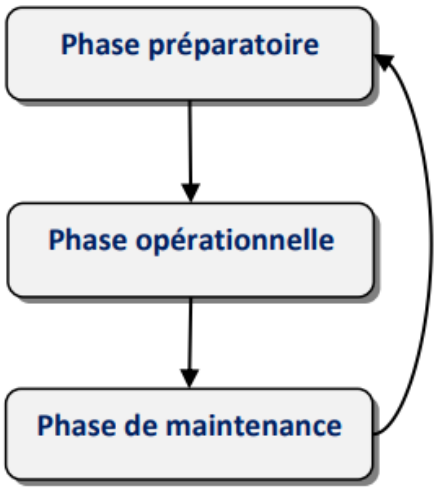


Figure I.2. Cycle de vie de l'objet.

- La phase préparatoire (bootstrapping) : déploiement des objets (capteurs, tags), leur configuration avec les informations nécessaires, par exemple les identificateurs, les clés de sécurité, etc
- La phase opérationnelle : dans la phase opérationnelle, l'objet connecté se met à réaliser sa mission qui diffère d'une application à une autre.
- La phase de maintenance : effectuer des mises à jours, régler les problèmes en faisant d'éventuelles réparations des objets en cas de défaillances par exemple. Il est même possible de remplacer carrément des objets et redémarrer à nouveau à partir de la phase préparatoire [5].

I.4. Technologies fondatrices de l'internet des objets

Bien que l'Internet des objets soit une notion relativement nouvelle, les technologies qui la rendent possible existaient depuis quelques années déjà. On parle alors des réseaux de capteurs sans fil et de la technologie d'identification par radio fréquence. Les évolutions observées par les technologies sans fil et le domaine des réseaux de télécommunication d'une part, et l'Internet de l'autre part, ont permis d'ouvrir de nouvelles perspectives pour ces technologies, qui ont pu s'instaurer efficacement dans notre vie quotidienne et qui sont devenues de plus en plus omniprésentes. Ainsi, de nouvelles facilités et de nouveaux modes d'exploitation des services peuvent être envisagés si les capteurs et les marqueurs d'identification intègrent l'Internet. Dans cette section nous présentons les technologies basiques et nous accentuons leurs rôles dans le contexte de l'Internet des objets.

I.4.1. L'identification par radio fréquence (RFID)

Un système RFID est composé d'un ou plusieurs lecteurs et d'un ensemble d'étiquettes (appelée aussi tags, marqueurs, identifiants ou transpondeurs) à micro-puissances. Les étiquettes sont des dispositifs minuscules équipées d'une puce contenant des informations et une antenne pour la communication radio. Elles sont placées sur les éléments que l'on veut identifier d'une manière unique ou tracer. Les étiquettes peuvent avoir différentes formes (Figure I.3) et peuvent être passives ou actives. Les étiquettes actives sont équipées d'une batterie, elles diffusent des signaux automatiquement et d'une façon autonome, tandis que les étiquettes passives ne disposent d'aucune source d'énergie et attendent à ce qu'un signal électromagnétique leur arrive et munit de l'énergie pour pouvoir envoyer leurs propres signaux. Les étiquettes passives sont plus déployées que celles qui sont actives car leur usage est beaucoup plus flexible avec un coût nettement réduit (comparé au coût relatif aux étiquettes actives qui est nettement

élevé). Une autre spécificité pas moins importante dans les étiquettes passives qui est la durée de vie. Par le fait d'être passive, la durée de vie de l'étiquette est importante (elle reste valable tant qu'elle garde son bon état), ce qui n'est pas le cas pour une étiquette active où la durée de vie est restreinte (s'achève avec l'épuisement de la batterie).

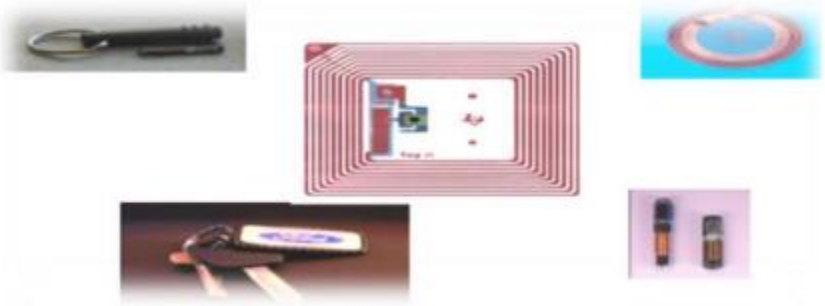


Figure I.3. Formes des étiquettes RFID.

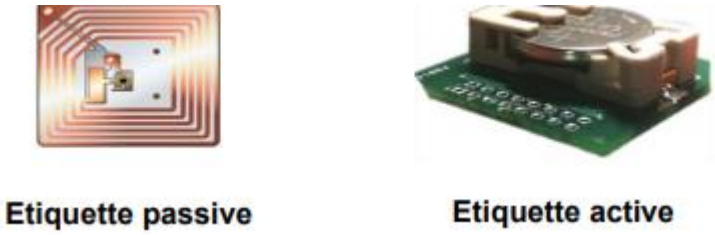


Figure I.4. Types des étiquettes RFID

Le processus d'identification se réalise à travers un scénario bien déterminé. En effet, le lecteur active les étiquettes qui passent devant lui en leur envoyant un signal électromagnétique puissant. Les étiquettent s'active et réagissent en répondant par un signal transportant les identités. Contrairement aux systèmes d'identification par codes barre qui exigent que le lecteur et le code barre soient exactement opposés et très proches l'un de l'autre, dans un système RFID, il suffit juste que le lecteur et l'étiquette soient l'un dans la portée de communication de l'autre pour que l'interaction puisse avoir lieu. La portée de

communication radio (appelée aussi la distance de lecture) dans un système RFID dépend du type de tag (passif ou actif) et de la gamme de fréquences utilisée. Par exemple, la portée avec les étiquettes actives est plus importante qu'avec celles qui sont passives. Dans le contexte de l'Internet des objets, les objets intelligents ont besoin d'être identifiés de façon unique. A partir de là, l'adoption de la technologie RFID s'est avérée nécessaire [6].

I.4.2. Les réseaux de capteurs sans fil

Les RCSFs jouent un rôle très intéressant dans l'Internet des objets. En effet, les capteurs permettent la représentation des caractéristiques dynamiques (température, humidité, pression, mouvements, …) des objets et des endroits du monde réel dans le monde virtuel représenté par le réseau Internet global. Ainsi, avec l'incorporation des réseaux de capteurs dans l'Internet, Les capteurs deviennent des serveurs (fournisseurs de services) dans ce que l'on désigne par le web des objets (dit WoT pour Web of Things). Ainsi, les services (applications) des RCSFs se rajoutent à l'ensemble des services et applications de l'Internet de futur qui réunira une variété de réseaux fortement hétérogènes (que ça soit sur le plan matériel ou logiciel), soumis à des contraintes différentes et qui sont déployés pour diverses applications, afin d'en avoir un monde réel très sophistiqué. En plus de ces deux technologies principales (RFID et RCSFs), on trouve également d'autres technologies qui contribuent à la concrétisation du principe de l'Internet des objets. On parle alors des systèmes embarqués et la nanotechnologie (rétrécissement et incorporation des capteurs et autres dispositifs miniatures dans les objets à faire connecter à Internet) [7] comme montre la figure suivante.

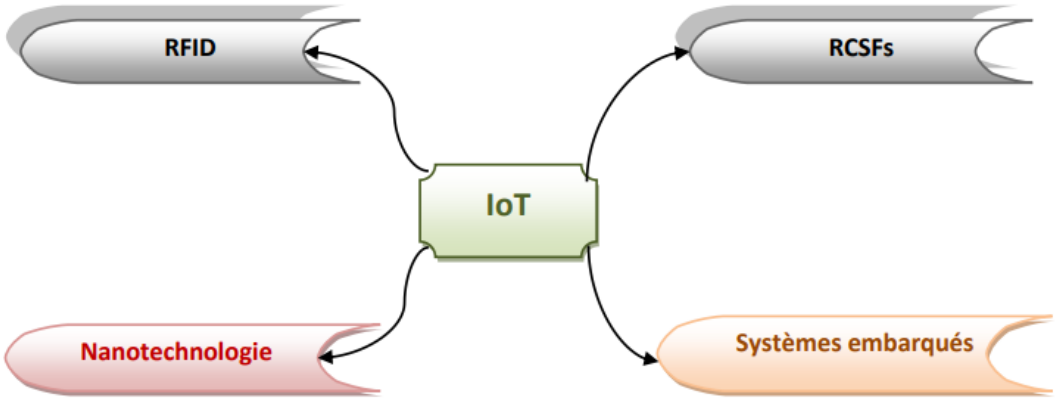


Figure I.5. Technologies fondatrices de l'Internet des objets

I.5. Architecture de l'Internet des objets

De point de vue architectural, on peut dire que l'Internet des objets est organisée en trois couches principales: la couche de perception de donnée, la couche réseau et troisièmement la couche application [8]. La figure ci-dessous illustre telle organisation.

I.5.1. La couche perception

La couche perception, au niveau bas dans la hiérarchie, est responsable de la capture de données, ainsi que leur identification dans leur environnement. Cette couche comprend ainsi le matériel nécessaire pour parvenir à la collection de données contextuelles des objets connectés, à savoir les capteurs, les étiquettes RFID, caméras, GPS (Global Positioning System), etc.

I.5.2. La couche réseau

Cette couche se charge de la transmission fiable des données générées dans la couche perception ainsi que l'assurance de la connectivité inter-objets connectés et entre objets intelligents et les autres hôtes de l'Internet. D'autre part, il est

prévu que les données issues de la couche perception soient énormes car le nombre d'objets connectés à Internet ne cesse d'augmenter à grands pas. De ce fait, il s'est avéré nécessaire de mettre en place des mécanismes et des équipements de stockage et de traitement massif de ces données sur Internet, à faible coût. Cela est bel et bien garanti par les services cloud qui assurent une gestion élastique des ressources de mémorisation et de traitement sur les géants centres de données résidant sur Internet et qui sont en mesure d'absorber efficacement la charge de données générée du côté de l'Internet des objets. à ce stade, il est important de noter que le cloud utilise un concept récent dénommé SDN (Software Defined Networking) qui vise une méthode de gestion abstraite basée sur le découplage des fonctionnalités décisionnelles et opérationnelles des équipements réseau, en vue de pouvoir déployer les tâches de contrôle sur des plateformes beaucoup plus performantes que les commutateurs classiques. Cela va réduire davantage la latence réseau et rendre possible l'automatisation de la gestion du large ensemble de serveurs sur le cloud et leur auto-configuration [9].

I.5.3. La couche application

Quant à elle, la couche application définit les profils des services intelligents et les mécanismes de gestion de données de différents types, provenant de différentes sources (différents types d'objets). Dans la section suivante nous abordons cet aspect applicatif et ce que représentent réellement les services intelligents dans chaque champ d'application. L'architecture peut être étendue à une quatrième couche dite la couche middleware entre la couche application et les deux autres couches. Cette couche sert pour une interface entre la couche matérielle et les applications. Elle comprend des fonctionnalités assez compliquées permettant la gestion des dispositifs, et traite aussi l'agrégation, l'analyse et le filtrage de données et le contrôle d'accès aux services. La couche middleware permet également la dissimulation de la complexité des mécanismes

de fonctionnement du réseau et rend plus facile le développement des applications par les concepteurs [10].

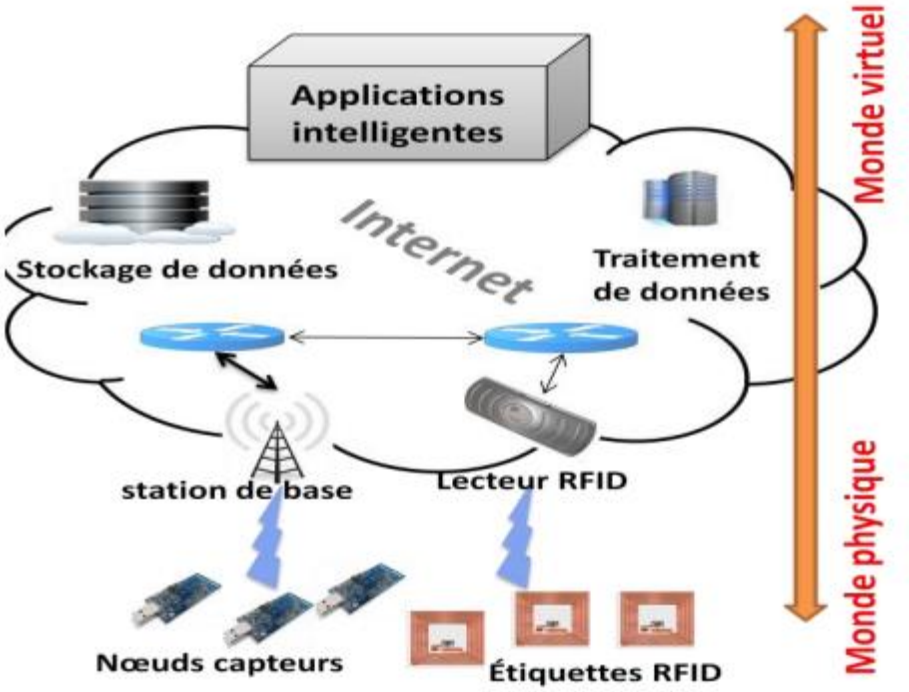


Figure I.6. Architecture de l'internet des objets.

I.6. Paradigmes de communication

En plus des communications humain à humain qui ont régné sur l'Internet classique, de nouveaux styles d'interactions émergent avec l'apparition de l'Internet des objets comme le montre la figure ci-dessous qui illustre ces interactions inter objets connectés et entre l'humain et le(s) objet(s) dans l'IoT.

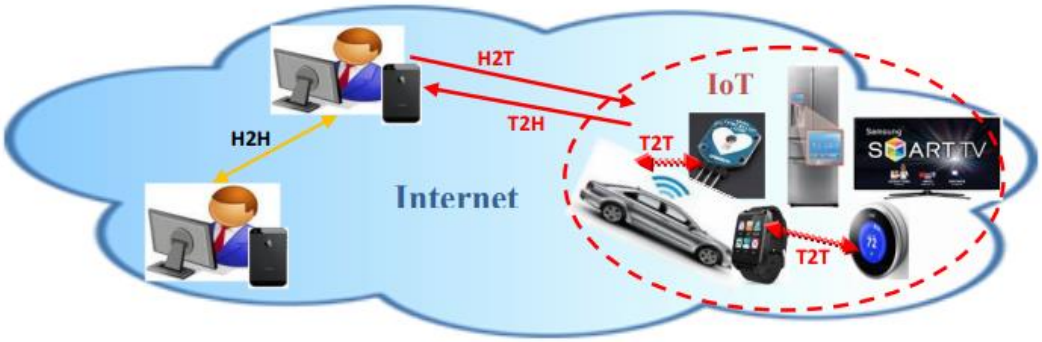


Figure I.7. L'émergence de nouveaux paradigmes de communication dans l'Internet du futur

I.7. Les communications IOT

I.7.1. Les communications humain-à-objet

L'utilisateur peut interroger des objets connectés à Internet à tout moment via son smartphone (ou autre dispositif connecté). Les communications humain-à-objet (dite aussi H2T pour Human-to-Thing) sont très fréquentes dans certaines applications de l'Internet des objets (Figure I.8) comme est le cas d'une application médicale ou de l'automatisation des maisons. Tel type d'interactions est caractérisé par une forte hétérogénéité matérielle et technologique car du côté de l'utilisateur on utilise généralement des équipements beaucoup plus puissants (ordinateur portable, Smartphone ou tablette) que les capteurs contraints du côté de l'objet sollicité dans l'IoT. Cependant, l'hétérogénéité dans toutes ses formes doit être traitée efficacement [11].

I.7.2. Les communications objet-à-objet

Les communications objet-à-objet (ou T2T pour Thing-to-Thing) sont appelées également machine-à-machine ou M2M (Machine-to-Machine). Cela désigne des communications automatiques et autonomes inter-machines sans l'intervention humaine. Rappelons que les communications M2M forment la base de l'informatique pervasive qui fait partie de l'ensemble des principes et concepts de l'Internet du futur. En fait, les interactions inter-objets intelligents dans l'IoT sont souvent homogènes, du moins au niveau des contraintes où on trouve des capteurs qui peuvent utiliser différentes technologies de transmission mais qui observent les mêmes limitations en termes de ressources et qui ont les mêmes vulnérabilités. Rappelons à ce stade que les communications allant des objets connectés dans l'Internet des objets vers les hôtes ordinaires de l'Internet (les communications objet-à-humain ou T2H: Thing-toHuman en anglais) sont aussi considérées comme une variante des communications M2M. Par exemple, un capteur associé à une porte d'une salle à accès restreint dans une banque, est configuré de telle sorte qu'il avise par MMS (ou e-mail) le responsable de la sécurité dans la banque (via son smartphone) en lui transmettant le temps

d'entrée de la personne ainsi que sa photographie. Cette opération se fait même si la personne était déjà authentifiée avant d'accéder la salle [12].

I.8. Les applications de l'Internet des objets

L'Internet des objets ce n'est pas qu'un immense ensemble d'objets intelligents interconnectés et connectés à Internet mais c'est également et plus considérablement, les applications qui sont en fait la raison d'être de cette nouvelle vague de connectivité sur Internet. L'existence des objets intelligents avec de nouvelles possibilités de communications automatiques et intelligentes vont sensiblement améliorer le mode de vie des gens ainsi que la qualité de services dans divers domaines à travers des degrés élevés d'autonomie et d'intelligence. Les potentialités de l'Internet des objets ont mené à ce que des modèles de nouvelles applications soient développées sur Internet. Dans cette section, nous citons les applications en vedette de l'IoT [13].

I.8.1. Les applications médicales

L'IoT aura de nombreuses applications dans le secteur de la santé où l'objectif est d'arriver à prévenir des situations graves et de suivre à distance des patients atteints des maladies chroniques et agir rapidement si cela s'est avéré nécessaire. Des capteurs corporels implantés dans le corps du patient récoltent des informations relatives aux paramètres médicaux, telles que la température, la glycémie, le rythme des battements du cœur ou encore même la tension artérielle. Ces informations seront stockées et traitées sur Internet (plus précisément sur un cloud) et mises à la disposition du médecin qui pourra les consulter n'importe quand et depuis n'importe quel dispositif connecté à Internet (ex : son Smartphone ou sa tablette). Le médecin est alerté en temps réel (en lui envoyant un mail ou un SMS) de tout changement brusque concernant l'état de son patient. Suivant le degré de gravité de la situation, le médecin réagit soit en se déplaçant chez le patient ou juste en le contactant et lui indiquant ce qu'il faut faire pour revenir à l'état normal. Imaginons par exemple un patient avec un

rythme cardiaque irrégulier. Le capteur détectant tel évènement déclenche une alerte au cardiologue s'occupant du patient. Le médecin peut également consulter à tout moment les rapports médicaux de ses patients ou bien interroger les capteurs pour avoir les valeurs actuelles.

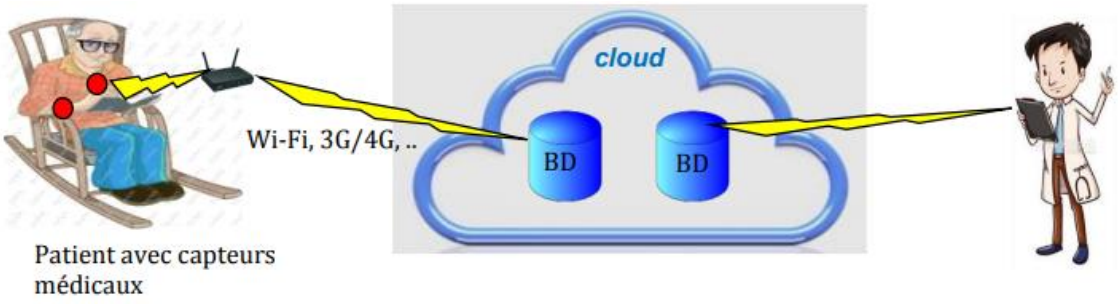


Figure I.8. L'internet des objets dans le domaine médical.

I.8.2. Les applications militaires

L'Internet des objets est un domaine fertile tant pour les applications civiles que pour les applications militaires. Dans le domaine de défense les capteurs et les nano-drones connectés à Internet permettent d'envisager des applications sophistiquées pour l'exploration, la surveillance des champs de batailles et des frontières, ainsi que la poursuite et la localisation géographique des objets connectés. Les forces militaires ont la tendance d'utiliser des infrastructures propriétaires pour la connectivité et les communications. En transitant vers l'Internet, il sera plutôt possible d'utiliser des infrastructures cloud, qui offrent une flexibilité opérationnelle très intéressante. Le soldat en mission peut lui-même être connecté à Internet à travers les capteurs connectés, intégrés dans sa tenue. Ces capteurs peuvent être par exemple des capteurs médicaux qui rapportent l'état de santé du soldat, ou des capteurs multimédia qui captent des images, une vidéo ou du son depuis la zone où il se trouve (le soldat).

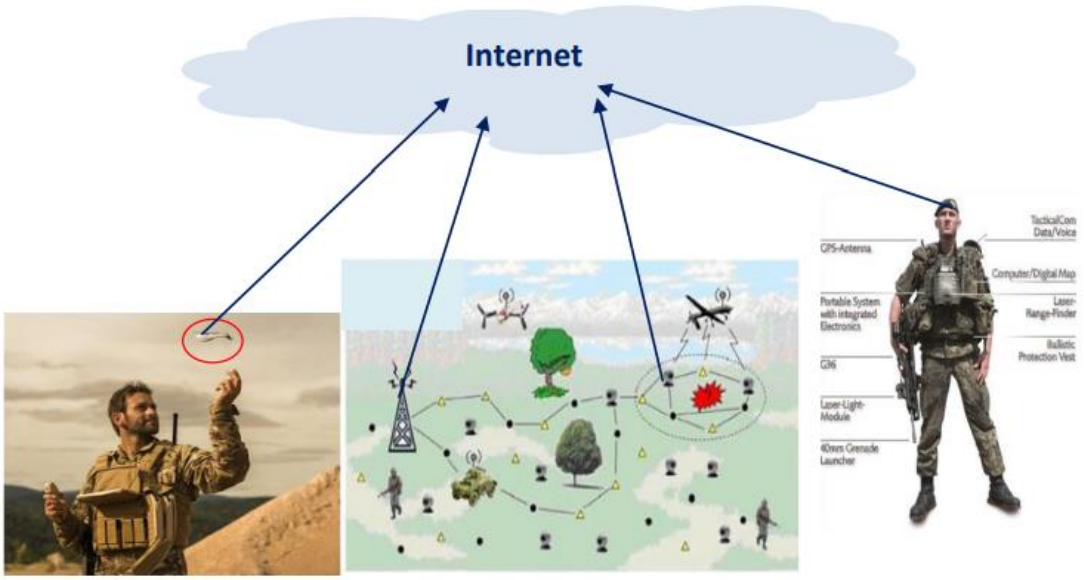


Figure I.9. Le domaine militaire et l'Internet des objets.

I.8.3. Les applications industrielles

Le secteur industriel est un autre domaine qui va être bouleversé par l'Internet des objets. Une quantité considérable de capteurs et d'étiquettes RFID et les contrôleurs embarqués, s'accroît sensiblement dans les systèmes de production industrielle, sur la chaîne logistique et même dans les produits. Ce qui aide les entreprises à améliorer la qualité de leurs processus de fabrication et à fournir un service après-vente plus concurrentiel. Ainsi, les usines connectées à Internet sont plus productifs, efficaces et intelligents que ceux qui ne le sont pas. En effet, le producteur peut également avoir une idée sur la commercialisation de ses produits à travers le monde, à l'aide des informations collectées auprès des différents points de ventes. D'autre part, les produits connectés seront capables de transmettre les avis (feedback) des clients aux producteurs pour faire un sondage sur le taux de satisfaction de la clientèle. Ils acquièrent et communiquent aussi des données pertinentes concernant les susceptibles pannes, les préférences des utilisateurs, ou autre. Il est important de noter que les

communications M2M jouent un rôle prépondérant dans l'automatisation des processus industriels et des interactions inter composants opérationnels.

I.8.4. Les maisons intelligentes

La maison du futur sera un objet connecté à Internet accessible à distance par ses propriétaires via des Smartphones, tablette ou ordinateurs connectés. La porte, la télévision, le thermostat, le réfrigérateur, les parapluies, les montres, etc. de telle sorte qu'une porte connectée informe les parents par Internet de la rentrée de leurs enfants. La télévision qui était seulement un terminal récepteur. Connectée à Internet, elle (la télévision) devient plutôt un dispositif émetteur/récepteur qui fournit à ses téléspectateurs la possibilité d'envoyer et recevoir des e-mails, faire des appels téléphoniques sur Internet, ou autre. Un thermostat intelligent connecté au réseau Wi-Fi de la maison permet de contrôler facilement la température de celle-ci à partir de n'importe où, pour une amélioration du confort et une optimisation des économies énergétiques. Le réfrigérateur intelligent connecté à Internet et muni d'un système RFID traque les produits élémentaires qui y sont stockés et enregistre des informations pertinentes leur concernant (comme la durée du stockage et la date d'expiration). L'utilisateur peut l'interrogé à distance pour savoir ce qui reste et ramener les produits manquant avant de rentrer à la maison. Ou alternativement, le réfrigérateur peut être programmé pour commander automatiquement les produits qui manquent.

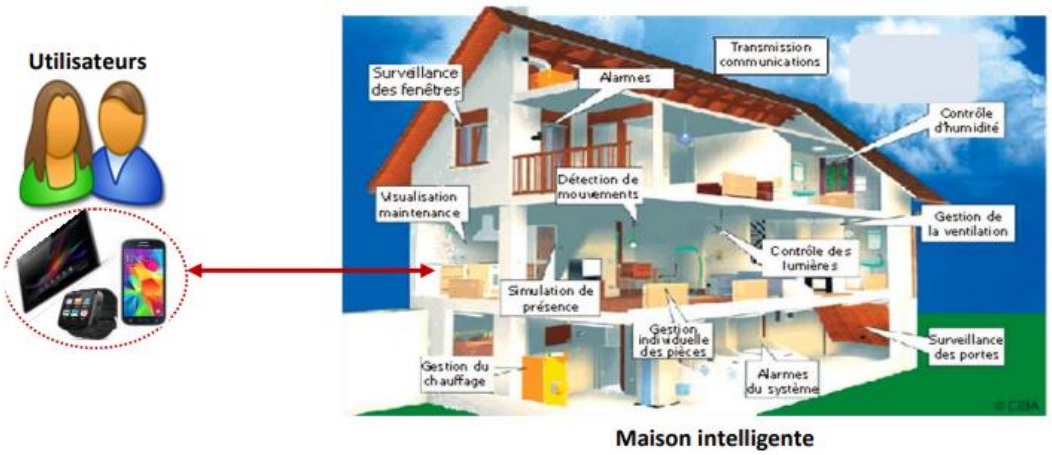


Figure I.10. L'Internet des objets et la domotique.

I.8.5. Les villes intelligentes

Pas que les maisons, les routes, les bâtiments, les véhicules, les magasins, les parkings, etc. seront tous connectés à Internet et annoncent leurs présence les uns aux autres objets connectés pour contrôler le trafic routier, aider les citoyens (surtout les automobilistes) à gagner le temps en leur fournissant des informations pertinentes, en temps réel, sur l'endroit où il se trouve (par exemple le plus proche parking, hôtel, restaurant, hôpital et autres) et des informations d'ordre général sur la ville, comme la température, le taux d'humidité les niveaux de radiation, … de même, les autorités de la ville intelligente trouveront une facilité de réalisation des tâches de contrôle de la pollution, l'éclairage urbain, etc. notons qu'une coexistence massive de multiples technologies est nécessaire pour la réalisation du projet de la ville intelligente, principalement les réseaux de capteurs.

Des applications avantageuses pas moins intéressantes peuvent être envisagées dans d'autres domaines à savoir l'agriculture de précision, où le principe est le même dans tous les cas: permettre un accès ubiquitaire aux informations relatives aux différents types d'objets intelligents existants dans notre environnement afin de parvenir à automatiser le contrôle et optimiser les rendements [10].

I.9. Les avantages de l'internet des objets

Nous avons cité dispersement dans différentes parties du présent chapitre quelques avantages de l'Internet des objets. Dans cette section, nous résumons les principaux avantages de l'IoT

- Accès ubiquitaire à l'information pour un monde plus intelligent et un mode vie sophistiqué et confortable.
- Amélioration de la qualité de service et de la télésurveillance dans différents domaines d'applications, à savoir le domaine industriel, médical, etc
- Améliorer la productivité et l'expérience-client : les objets connectés envoient des rapports à leurs constructeurs indiquant les préférences et les habitudes des clients aidant davantage les entreprises à agir de manière proactive et adaptée qui satisfait la demande et les exigences de la clientèle.
- Le gain du temps est un autre avantage de l'IoT. Les déplacements inutiles sont dès lors remplacés par une simple navigation sur le web pour commander des produits, contrôler l'état des objets et/ou endroits connectés.

- Dans certaines applications, l'IoT nous permet même de rationaliser nos dépenses et faire des économies car on ne consomme qu'en cas de besoin, que ça soit pour les achats ou la consommation énergétique (nécessaire pour l'éclairage ou la climatisation) ou autre
- Possibilité d'exploitation des ressources géantes de l'Internet pour le stockage et le traitement des données écoulées de l'IoT.

I.10. Les enjeux de l'Internet des objets

Bien que l'Internet des objets soit un concept qui est à la fois avantageux et prometteur, et qui pourra apporter des solutions efficaces des problèmes du suivi et de télésurveillance dans différents domaines. En contrepartie, l'IoT soulève certaines questions décisives, étroitement liées à sa maturité et son acceptabilité. On cite ci-dessous les enjeux les plus marquants.

La sécurité : la sécurité des personnes, des communications, des données, des services, des réseaux et des équipements était et continue à être un problème sévère observé par l'internet courant. Aujourd'hui avec la naissance de l'IoT, l'amplitude du problème va prendre un autre ordre de gravité. Des milliers d'objets contraints connectés en permanence à internet et intégrés dans toute sorte d'objets dans notre vie quotidienne, vont porter le risque d'être ciblés par les menaces classique de l'Internet. Il est même possible que de nouvelles générations d'attaques apparaissent. Donc, les objets intelligents dans l'IoT, la transmission et le stockage de leurs données sur Internet devraient être sécurisés. D'autre part, l'IoT peut lui-même menacer la sécurité des individus ou des institutions. L'armée chinoise proscrit les officiers et les soldats de porter des objets connectés (comme les montres et les lunettes connectées à Internet) et considère leur utilisation comme une violation de la réglementation sur le secret dans les casernes [14].

La protection de la vie privée des utilisateurs : un grand nombre de capteurs connectés à Internet et intégrés dans des objets d'usage quotidien révèlent nos habitudes notre état de santé notre localisation géographique et autres types d'informations qui nous sont privées. Il devra absolument y avoir des mécanismes robustes qui peuvent assurer la confidentialité des données que l'utilisateur qualifie être sensibles. Les utilisateurs devraient également pouvoir savoir qui accède quelles données (concernant les utilisateurs) sur Internet et pour quelle raison.

Les limitations de ressources : les capteurs et les tags RFID sont très limités en ressources de calculs, de stockage mémoire et d'énergie. A cet effet, les solutions (protocoles de communications ou de sécurité, technologies de transmission, etc.) destinées à l'Internet des objets doivent prendre en considération telles contraintes et limitations.

L'hétérogénéité : des dispositifs de divers types ayant des capacités variées et appartenant à des réseaux de différentes natures, vont intégrer l'Internet en utilisant différentes technologies de communication (filaire, sans fil, satellitaire). Avec toutes ces formes d'hétérogénéités matérielles et technologiques, il serait primordial de mettre en place des mécanismes bien avertis qui soient capables d'en cacher et gérer.

L'interopérabilité : c'est parmi les plus grands défis de la réalisation de l'Internet des objets. L'interopérabilité c'est, en réalité, la cohabitation des dispositifs, des systèmes et des mécanismes disjoints et la possibilité de les faire coopérer et interagir en toute flexibilité. Une tendance récente tend vers la standardisation et l'unification des systèmes et protocoles opérationnels dans l'IoT et de les présenter en open source (à accès libre). Ceci afin de faciliter la collaboration entre objets connectés, ainsi que le couplage avec les entités externes se trouvant sur Internet.

La virtualisation : plusieurs capteurs connectés peuvent représenter un seul capteur virtuel qui rapporte une mesure virtuelle résultant de l'agrégation de

plusieurs états secondaires. Par exemple un capteur virtuel qui nous dit si l'état de santé du patient est bon ou non. Cette information n'est qu'une combinaison de plusieurs informations fournies par plusieurs capteurs médicaux réels incorporés dans le corps du patient. Ainsi, un model générique de virtualisation des objets connectés à l'IoT, nommé VoT (Virtualization of Things) [15] permet une représentation abstraite des objets et l'accumulation des données qui en proviennent, depuis différents endroits, pour faciliter leur contrôle.

La transparence : l'objectif de l'informatique transparente est de rendre les systèmes informatiques des boites noires transparentes à travers des communications sans fil, automatiques et invisibles ne nécessitant pas l'interaction avec les utilisateurs. La transparence est la base de l'informatique pervasive qui est à son tour un facteur essentiel dans l'Internet des objets.

Le nombre croissant d'objets connectés : il est prévu que le nombre d'objets intelligents qui vont peupler l'Internet du futur franchira les millions, voir les milliards. Avec cela, l'adoption de nouveaux mécanismes qui supportent efficacement l'évolutivité continue dans le nombre d'objets connectés, est vivement recommandée.

La mobilité : un nombre immense d'objets connectés à Internet en tant que partie de l'Internet des objets, seront le plus souvent mobiles. De ce fait, des solutions flexibles de gestion de la mobilité doivent être mises en place pour permettre à tels objets d'accomplir leurs missions efficacement indépendamment de la fréquence et la vitesse de la mobilité.

La qualité de service des communications : suivant que l'application est critique ou non, les communications inter objets connectés dan l'IoT et entre ces derniers et les hôtes ordinaires de l'internet, peuvent exiger ou non un minimum de qualité de service en termes de délais, débits, fiabilité, etc

CONCLUSION :

L'Internet des objets en tant qu'une évolution de l'Internet actuel permet une amélioration considérable de notre mode de vie et la façon dont les objets intelligents dans notre entourage interagissent entre eux et avec leurs utilisateurs de telle sorte que nos activités, nos biens, notre état de santé, nos dépenses,…puissent être contrôlés efficacement et d'une manière ubiquitaire.

Dans ce chapitre, nous avons discuté principalement les technologies de base ainsi que les applications en vedette de l'IoT. Nous avons aussi mis en évidence les contraintes liées au déploiement de l'IoT et qui devraient être soigneusement traitées pour atteindre les objectifs prédéfinis et parvenir à optimiser les rendements.

Chapitre II :
Sécurité Informatique dans l'IOT

INTRODUCTION

Avec le développement de l'utilisation d'internet, de plus en plus d'entreprises ouvrent leur système d'information à leurs partenaires ou leurs fournisseurs, il est donc essentiel de connaître les ressources de l'entreprise à protéger et de maîtriser le contrôle d'accès et les droits des utilisateurs du système d'information. Il en va de même lors de l'ouverture de l'accès de l'entreprise sur internet. Par ailleurs, avec le nomadisme, consistant à permettre aux personnels de se connecter au système d'information à partir de n'importe quel endroit, les personnels sont amenés à « transporter » une partie du système d'information hors de l'infrastructure sécurisée de l'entreprise.

Ce chapitre est consacré à la sécurité de L'IOT envers les vulnérabilités et menaces dans ces objets connectés.

II.1. Vulnérabilités et menaces dans l'internet des Objets

A cause de la forte intégration de l'IOT, les objets du quotidien deviennent des risques potentiels d'attaque sur la sécurité, l'ubiquité de L'IoT amplifiera les menaces classiques de la sécurité qui pèsent sur les données et les réseaux, de plus l'apparition de nouvelles menaces qui toucheront directement à l'intégrité des objets eux-mêmes, les infrastructures et processus et la privacy des personnes [16].

II.1.1. Menaces sur les données et les réseaux

Le manque de surveillance et de protection physique des objets communicants peut en gendres des attaques potentielles portées sur le matérielle telles que le vol, la corruption ou la contrefaçon de ces derniers pour récupération des données qui sont stockées sur ces dispositifs ou pour interrompre le bon fonctionnement des réseaux ou les systèmes complexes les hébergent. De plus, les transmissions sans fil sont réputées par leur forte vulnérabilité aux attaques

de l'écoute passive et de déni de service. Les solutions cryptographiques existantes aujourd'hui ne sont pas adéquates pour tenir faces à ces problèmes cités à cause de la limitation de ressources des objets communicants, de ce fait, l'adaptation de ces dernières ou la conception de nouveaux modèles est une nécessité afin d'assurer les services de sécurité [17].

II.1.2. Menaces sur la vie privée

De nombreux objets seront intégrés, portés ou même bien installés dans les lieux privés des personnes, ces objets présentent une potentielle menace pour la vie privée (privacy) de leurs utilisateurs. En effet, ces appareils électrique non seulement sont traçables, mais peuvent filmer, écouter ou même enregistrer leurs rythmes cardiaque ou respiratoire ainsi que la température du corps ou sa cinématique dans le but d'un malicieux [16].

II.1.3. Menaces sur les systèmes et l'environnement physique des objets

Des objets malicieux connectés à un réseau ou intégrés dans un système complexe peuvent causer un dysfonctionnement quelconque,, un déni de service ou autres types d'attaques à l'intégrité des données et les informations sensibles du système, ou pire encore prendre le contrôle du système en causant des importants [16].

II.2. La sécurité dans internet des objets

Définition des objectifs de la sécurité

La sécurité informatique d'une manière générale consiste à assurer que les ressources matérielles et logicielles d'une organisation sont uniquement dans le cadre prévu. Elle vise à assurer plusieurs objectifs, dont les cinq principaux sont L'authentification, L'intégrité, la disponibilité et la non-répudiation [17], [18].

II.2.1. Authentification

L'authentification peut être définie comme le processus de prouver une identité revendiquée. La confidentialité, l'intégrité des données, et la répudiation dépendent tous de l'authentification appropriée. Un système sans cette fonctionnalité ne pouvait pas fournir les objectifs de sécurité mentionnée de manière satisfaisante .

II.2.2. Confidentialité

Ensemble des mécanismes permettant qu'une communication de données reste privée entre un émetteur et un destinataire. La cryptographie ou le chiffrement des données est la seule solution fiable pour assurer la confidentialité des données [18].

II.2.3. Intégrité

L'intégrité peut être vue comme un ensemble de mesures garantissant la protection des données contre les modifications et les altérations non autorisées. L'objectif des attaques sur l'intégrité est de changer, D'ajouter ou supprimer des informations ou des ressources .

II.2.4. Disponibilité

La disponibilité est un service réseau qui donne une assurance aux entités autorisées d'accéder aux ressources réseaux avec une qualité de service adéquate. L'objectif des attaques sur la disponibilité est rendre le système inexploitable ou inutilisable.

II.2.5. Non-répudiation

Mécanisme permettant de garantir qu'un message a bien été envoyé par un émetteur et reçu par un destinataire, c'est-à-dire au des correspondants ne pourra nier l'envoi ou la réception du message [16], [18].

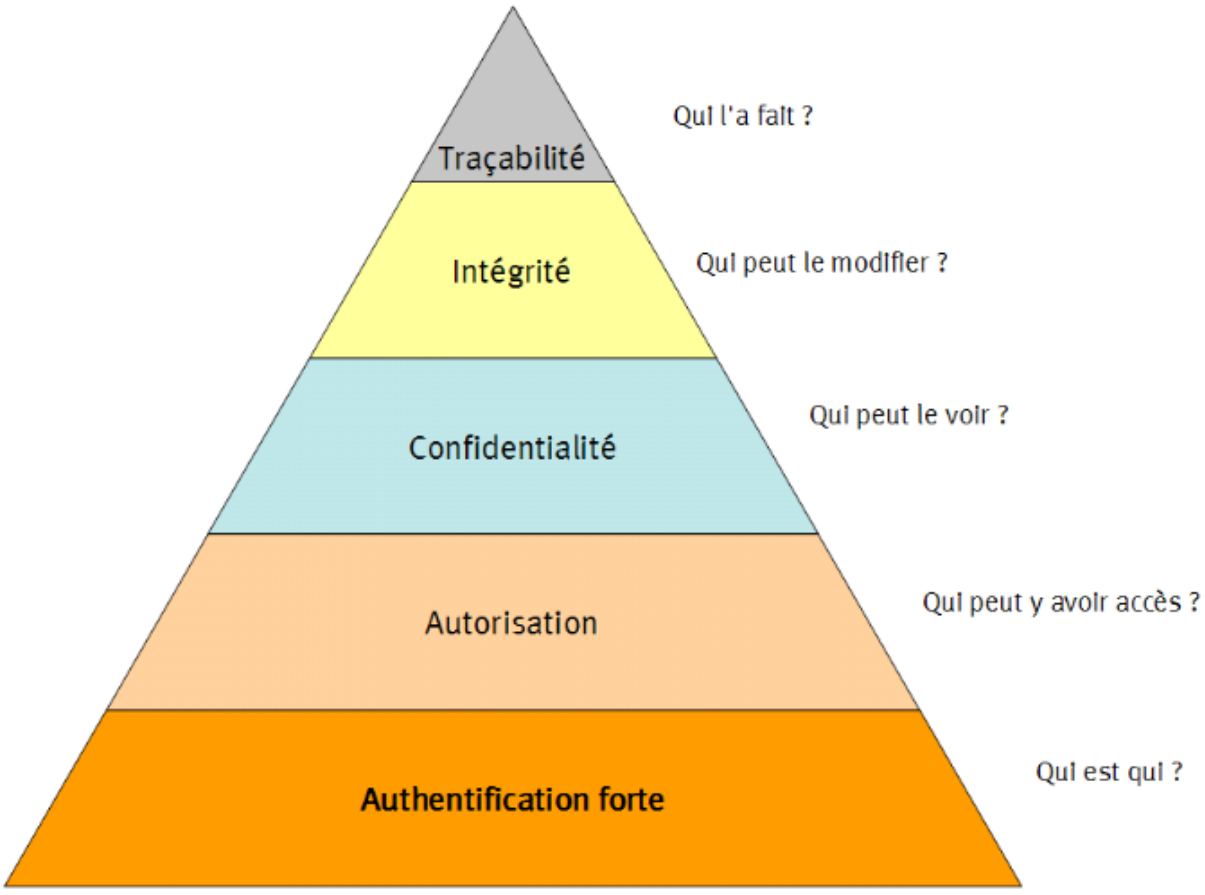


Figure II.1. Objectifs de la sécurité.

II.3. Contexte sécuritaire de l'IOT

II.3.1. Un manque de sécurité flagrant

Les objets de l'IoT, en raison du nombre important d'appareils et de la variété des domaines impactée, nécessitent une sécurisation appropriée. Le cryptologie Bruce Schneier décrit en 2016 dans un article de blog que "le marché récompense encore largement le sacrifice de la sécurité au profit du prix et du délai de mise sur le marché" [19]. Cette constatation est confirmée par Barcena et Wueest [20] qui, dans leur étude sur 50 des objets les plus commercialises, ont constaté qu'aucun de ces objets n'imposait de mots de passe suffisamment robustes, n'utilisait d'authentification mutuelle entre clients/serveurs ou ne

bénéficiait de protection contre les attaques par force brute [20].

II.3.2. Une nouvelle surface d'attaque

La prolifération des objets connectes au sein de multiples domaines d'application (illustrées en figure II.2) augmente le nombre de ressources dont disposaient initialement les systèmes d'information. Ces ressources représentent de nouveaux vecteurs d'attaques sur les objets connectes, mais également sur des services plus critiques y ´étant relies comme le montrent Stellios et al. dans leur études [21]. Les nouvelles surfaces d'attaque introduites par l'IoT peuvent être exploitées afin de pénétrer dans une infrastructure informatique sécurisée et extraire de l'information tel qu'effectue lors de l'attaque d'un casino via un capteur présent dans un aquarium en 2017 [17]. La figure II.3 illustre ces propos en représentant l'apparition d'un nouveau vecteur d'attaque cause par l'ajout d'une cafetière connectée, accessible depuis une application mobile, sur une structure (a priori) sécurisée. Hossain et al. Proposent, dans leur papier, un scenario dans lequel un seul objet compromis au sein d'un réseau interconnecte peut compromettre l'ensemble du réseau [22].

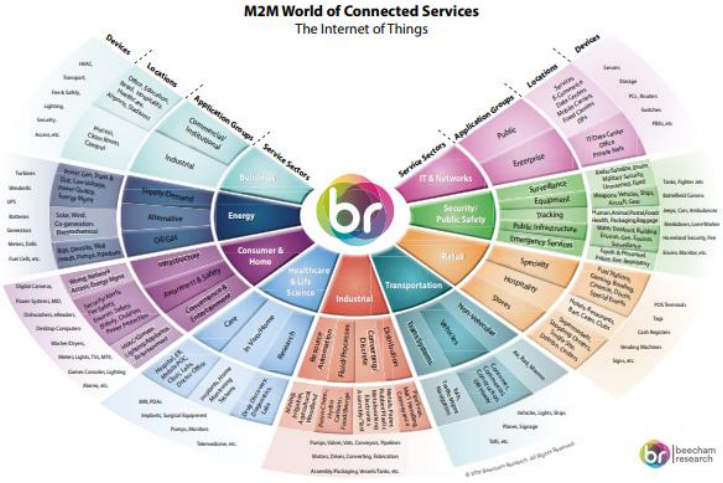


Figure II.2. Diversité des domaines d'application de l'Internet des Objets en 2017 (source : Beecham Research)

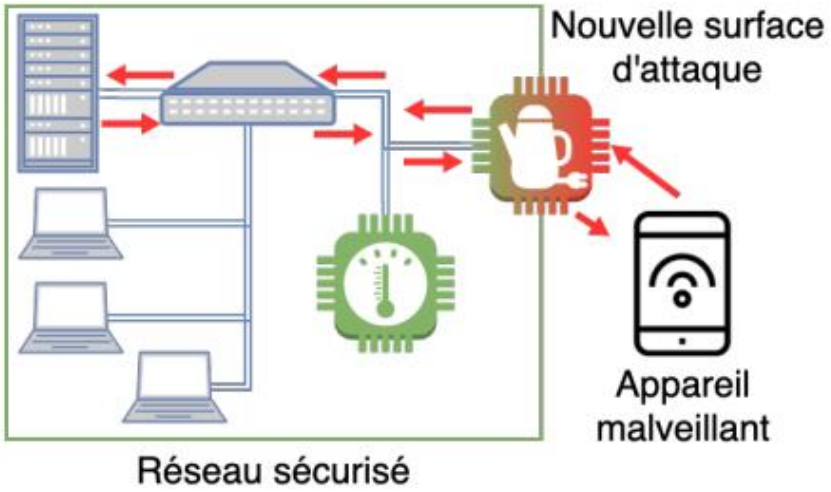


Figure II.3. Nouvelle surface d'attaque suite à l'ajout d'un objet connecte au sein d'une infrastructure sécurisée

II.4. Quelques attaques dans l'IOT

A l'instar des Services d'Information (SI) classiques, l'Internet of Things n'est pas exempt d'attaques. L'OWASP propose un top 10 des pratiques a éviter lors de la conception d'applications IoT [23]. Ces pratiques vont des mots de passe faibles ou encodes en dur dans l'appareil au manque de mesures de sécurisations matérielles en passant par l'absence de mécanisme de mise à jour sécurise. Quand une application IoT n'est pas conçue afin d'éviter ces pratiques, elle se retrouve susceptible de subir une attaque

Barcena et Wueest décrivent dans leur papier deux attaques utilisant des méthodes d'attaques sur SI classiques [22]. Une première consiste a effectuer une injection SQL auprès de l'interface cloud d'une application connectee offrant ainsi la possibilite `a l'utilisateur de déconnecter l'ensemble des objets du service ou de s'emparer des données d'identification. La seconde attaque décrite consiste à effectuer une attaque Man-In-The-Middle lors de la

vérification des mises à jour d'un hub intelligent. Pour ce faire, les auteurs effectuent du ARP poisoning pour se faire passer pour le serveur de mise à jour de l'objet. Le paquet de mise à jour n'étant pas chiffré ni signé, les auteurs peuvent ainsi l'altérer et le fournir à l'appareil lors de sa requête de mise `a jour. Similairement, l'attaque de Target en 2014 repose sur une négligence dans l'élaboration du réseau interne de l'entreprise [19]. Dans cette attaque, le système permettant de contrôler le chauffage, la ventilation et la climatisation (HVAC) était directement relie avec le terminal bénéficiant d'un accès aux données bancaires des clients. L'attaquant a pénètre le réseau de Target après avoir subtilise des identifiants d'accès `a l'entreprise de maintenance du système HVAC. Malgré le fort impact médiatique de l'attaque de Target, ce n'est rien en comparaison de celui du botnet Mirai [20, 22]. Ce botnet exploite une multitude de périphériques IoT dont la plus grande proportion correspond `a des webcams avec des identifiants de connexions faibles ou par défaut. Les périphériques alors infectes agissent alors comme des zombies et suivent les ordres de l'attaquant en envoyant une multitude de requetés auprès des serveurs d'OVH et du site web de KrebsOnSecurity [22].

II.5. Vie privée dans l'IOT

La protection des données personnelles est une opération difficile dans le domaine de l'IOT. En effet, la quantité colossale d'informations couplée avec leurs natures variées et les besoins, parfois de traitement en temps réel, ne facilitent pas cette procédure. En général, les données sont soumises aux 3 étapes de traitement suivantes : collecte, agrégation et analyse (ou data mining) [26]. La Figure **II.4** illustre ces étapes et les met en relation avec leurs architectures.

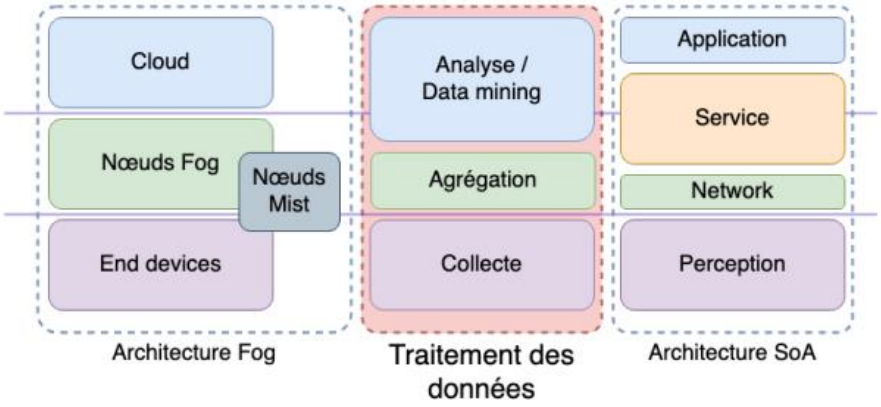


Figure II.4. Mise en relation des étapes de traitement des données avec les architectures fog et SoA.

- Le Fog computing est une plateforme hautement virtualisée qui fournit des services traditionnellement présents dans le cloud computing (traitement de données, stockage, connectivite, etc.), généralement, mais pas exclusivement, en périphérie du réseau [27], [25].
- (SoA) Cette architecture reprend la structure IOT en 3 couches et introduit une quatrième couche intitulée Service entre Network et Application comme illustrée en Figure II.1 [30], [31], [32].

II.6. Classification selon la cible d'attaque:

II.6.1. Les attaques réseaux :

Les attaques réseaux s'appuient sur des vulnérabilités liées directement aux protocoles ou à leur implémentation. Il existe un grand nombre. Néanmoins, la plupart d'entre elles ne sont que des variantes des cinq attaques réseaux les plus connues aujourd'hui.

II.6.2. Les attaques applicatives :

Les attaques applicatives s'appuient principalement sur des vulnérabilités

spécifiques aux applications utilisées.

II.7. Exemples d'attaques

L'essor de l'IoT a fait des heureux : les cybercriminels. De plus en plus de hackers s'appuient sur les failles de sécurité des objets connectés pour créer un botnet et mener une attaque à grande échelle [35].

II.7.1. Le Botnet

Le terme botnet est la contraction des mots robot et network (réseau), il désigne un réseau de bots informatiques, autrement dit un ensemble de programmes connectés au web et communicant entre eux – ou avec d'autres programmes – afin d'effectuer certaines tâches, généralement malveillantes comme le montre la figure II.5. Beaucoup les emploient en effet pour inonder le web de spams, pour mener des opérations d'hameçonnage ou encore pour participer à des attaques de déni de service, que l'on nomme DDoS (voir plus bas). Pour d'autres, le botnet peut également avoir vocation à créer une fraude au clic, à trouver des mots de passe ou encore à miner des cryptomonnaies. [36].

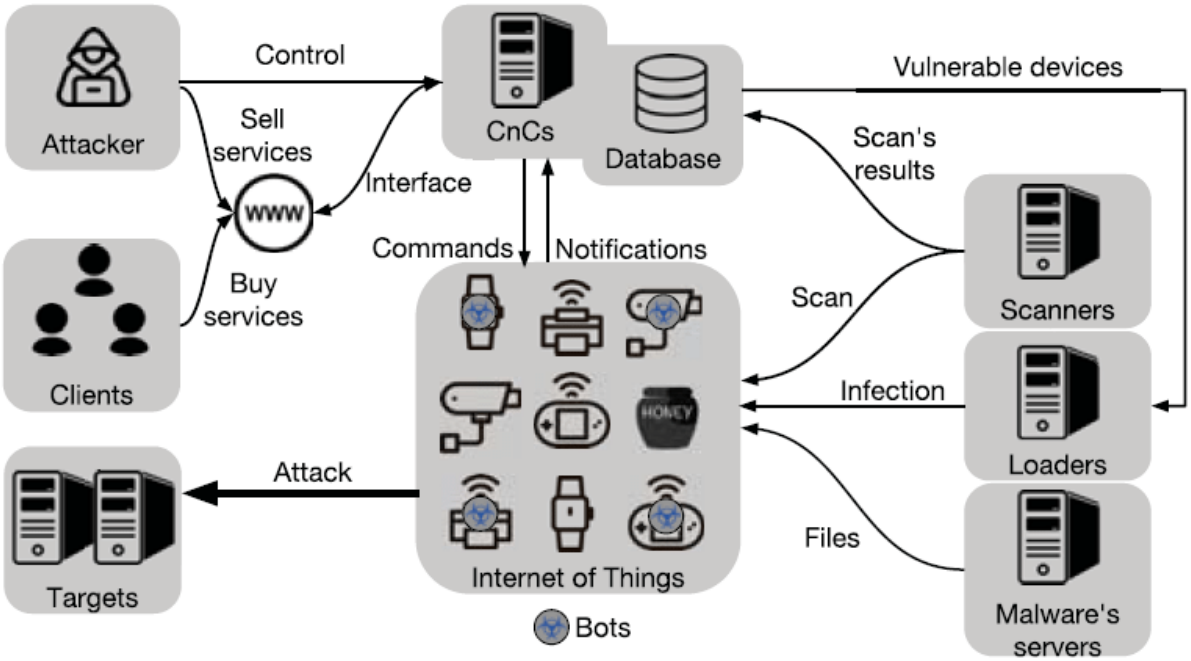


Figure II.5. Aperçu du Botnet en IOT

II.7.2. Botnet DDoS

Les attaques par déni de service distribuées (ou DDoS) définissent des attaques informatiques dont le but principal consiste à empêcher le bon fonctionnement d'un service. Dans la pratique, les attaques DDoS peuvent être menées pour empêcher l'accès des utilisateurs à un serveur web, pour rendre impossible la distribution de mails au sein d'une entreprise, etc. Avec le boom des objets connectés, les individus malveillants disposent de nouvelles armes pour mener leurs attaques DDoS. Il leur "suffit" en effet de prendre le contrôle de ces objets via les failles de leur système de sécurité pour déclencher ensuite une attaque de grande ampleur via un botnet.

La Figure II.6 montre le scénario d'attaque DDoS sur un contrat intelligent dans les systèmes blockchain-IoT

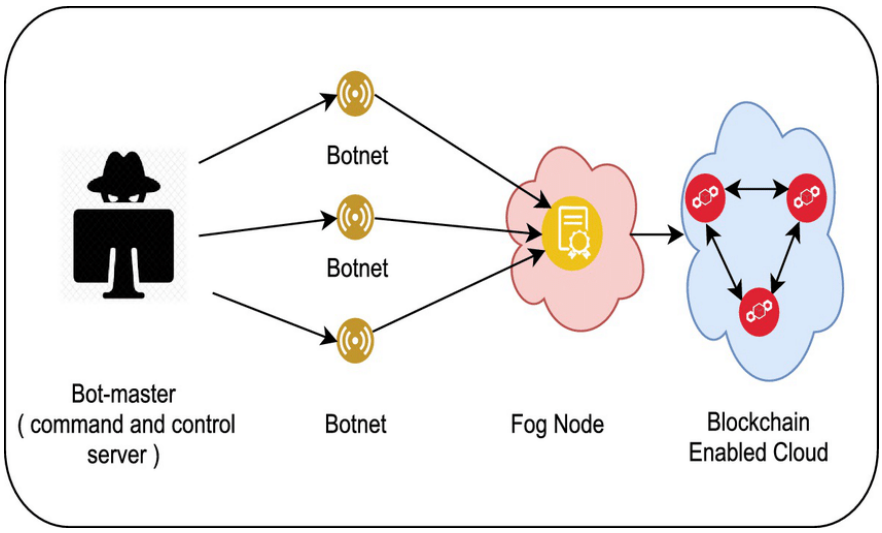


Figure II.6. Scénario d'attaque DDoS sur un contrat intelligent dans les systèmes blockchain-IoT

II.7.3. Botnet Mirai

En 2016, toute la presse spécialisée et une partie de la presse généraliste ne parlaient que de ça : le botnet Mirai. Ce logiciel malveillant capable de transformer des ordinateurs fonctionnant sous Linux en bots contrôlés à distance est à l'origine d'une cyber-attaque à très grande échelle. À l'automne 2016, on découvre en effet qu'un ou plusieurs botnets Mirai ont été utilisés pour lancer l'une des plus importantes attaques DDoS. Les cibles retenues : le site de sécurité informatique du journaliste Brian Krebs, l'hébergeur français de sites web OVH puis la société Dyn. Cette dernière attaque a d'ailleurs paralysé pendant plus d'une dizaine d'heures de nombreux sites et services, tels que Twitter, PayPal, AirBnB ou encore Netflix. Les experts sont parvenus à identifier le fonctionnement de Mirai : celui-ci repose sur la recherche permanente sur Internet des adresses IP qui correspondent à des objets connectés (IoT). Après avoir identifié les objets connectés vulnérables, Mirai s'y connectait pour y installer le logiciel malveillant. Mirai reste en 2021 le logiciel malveillant le plus courant dans les attaques IoT.

La figure II.7 montre une attaque de botnet DDoS est assez simple. Il donne des commandes au serveur de contrôle. Et le serveur de contrôle envoie des commandes d'attaque à chacun des nœuds individuels (appareils infectés) du botnet. Ils envoient à leur tour le trafic d'attaque à la cible [37].

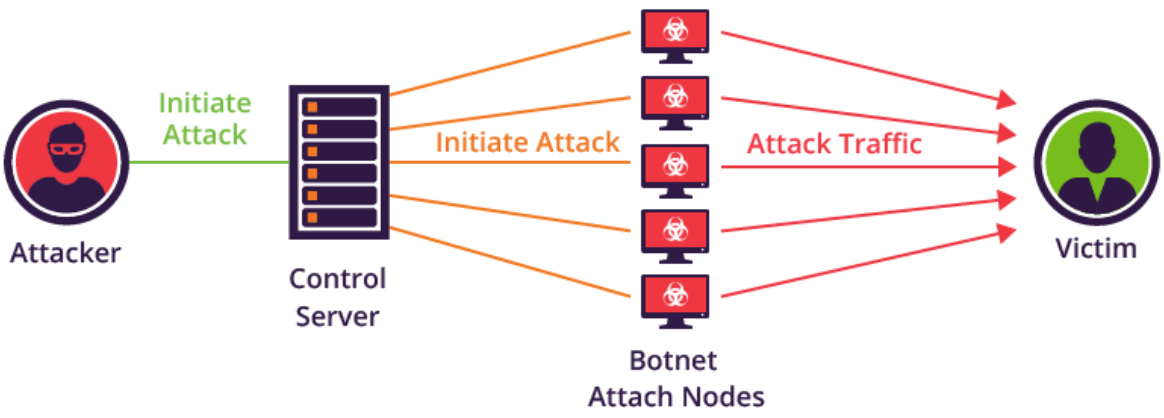


Figure II.7. Attaque de botnet DDoS

II.8. Mécanismes de défense contre les attaques :

C'est l'ensemble de procédures ou dispositifs qui sont conçu pour détecter, prévenir ou récupérer les attaques qui menacent la sécurité informatique, il existe plusieurs outils de prévention contre-attaques informatiques, Nous avons cité ci-dessous quelques mécanismes [38].

II.8.1. Chiffrement :

Algorithme généralement basé sur des clefs et transformant les données. Sécurité dépendante du niveau de sécurité des clefs, la figure **II.8.** explique tout

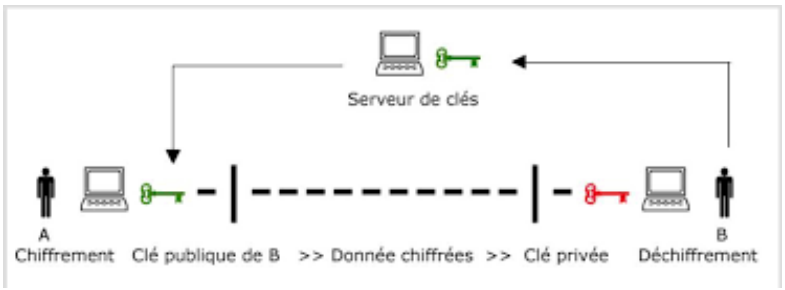


Figure II.8. Chiffrement

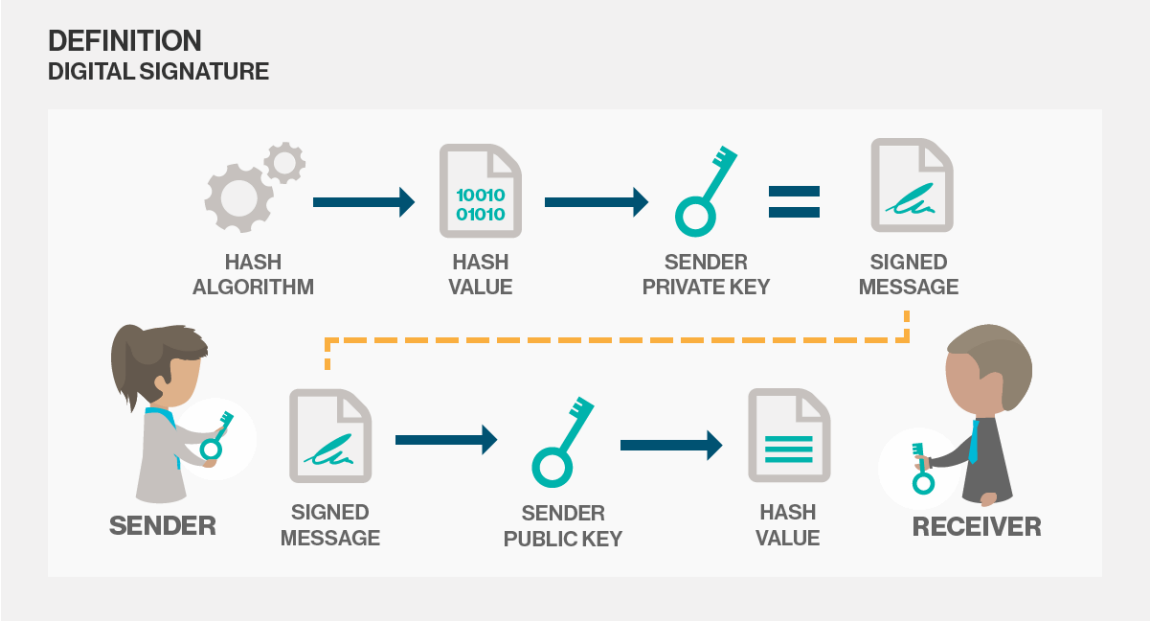


Figure II.9. Signature numérique

II.8.2. Signature numérique: Données ajoutées pour vérifier l'intégrité ou l'origine des données.

II.8.3. Bourrage de trafic : Données ajoutées pour assurer la confidentialité, notamment au niveau du volume du trafic.

II.8.4. Chevaux de Troie: sont des programmes qui en plus d'une fonction classique, ont une fonction cachée nuisible, récupérer vos mots de passe, détruire le disque dur, etc.

II.8.5. Notarisation : Utilisation d'un tiers de confiance pour assurer certains services de sécurité.

II.8.6. Contrôle d'accès : Vérifie les droits d'accès d'un acteur aux données.

II.8.7. Antivirus : Logiciel censé protéger l'ordinateur contre les logiciels (ou fichiers potentiellement exécutables) néfastes. Ne protège pas contre un intrus qui emploie un logiciel légitime, ou contre un utilisateur légitime qui accède à une ressource alors qu'il n'est pas autorisé à le faire.

II.8.8. Le pare-feu : Un élément (logiciel ou matériel) du réseau informatique contrôlant les communications qui le travers. Il a pour fonction de faire respecter la politique de sécurité du réseau, celle-ci définissant quelles sont les communications autorisées ou interdites. N'empêche pas un attaquant d'utiliser une connexion autorisée pour attaquer le système. Ne protège pas contre une attaque venant du réseau intérieur (qui ne le traverse pas).

II.8.9. Détection d'intrusion : Repère les activités anormales ou suspectes sur le réseau surveillé. Ne détecte pas les accès incorrects mais autorisés par un utilisateur légitime. Mauvaise détection : taux de faux positifs, faux négatifs.

II.8.10. Journalisation ("logs") : Enregistrement des activités de chaque acteur. Permet de constater que des attaques ont eu lieu, de les analyser et potentiellement de faire en sorte qu'elles ne se Reproduisent pas.

II.8.11. Analyse des vulnérabilités ("Security audit") : Identification des points de vulnérabilité du système. Ne détecte pas les attaques ayant déjà eu

lieu, ou lorsqu'elles auront lieu. Mais Aucun des mécanismes de sécurité ne suffit par lui-même, et pour cela dans la plupart du temps en vue d'atteindre un niveau acceptable de sécurité informatique plusieurs mécanismes sont utilisés en même temps [39].

CONCLUSION

A travers ce chapitre, nous avons présenté des généralités et les différents objectifs de la sécurité informatique dans le domaine de l'internet des objets. Nous avons accentué les défis et les principaux aspects liés à la sécurisation des communications dans tels réseaux contraints contre toute manipulation mal intentionnée ou attaque qui touche la vie privée dans IOT. Finalement, nous avons présenté les mécanismes de défense contre ces attaques ou principalement la cryptographie dont on va se focaliser bien évidemment dans le prochain chapitre.

Chapitre III :

Cryptographie légère pour l'internet des objets

INTRODUCTION

L'origine de la cryptologie mot réside dans la Grèce antique. La cryptologie est un mot composé de deux éléments : «cryptos », qui signifie caché et « logos» qui signifie mot. La cryptologie est aussi vieille que l'écriture elle-même, et a été utilisé depuis des milliers d'années pour assurer les communications militaires et diplomatiques, par exemple, le célèbre empereur romain Jule César utilisait un algorithme de chiffrement pour protéger les messages à ses troupes. Dans le domaine de l'un de cryptologie peut voir deux visions : la cryptographie et la cryptanalyse. Le cryptographe cherche des méthodes pour assurer la sûreté et la sécurité des conversations alors que le Crypto analyse tente de défaire le travail ancien en brisant ses systèmes. La cryptographie traditionnelle est l'étude des méthodes permettant de transmettre des données de manière confidentielle et la cryptanalyse, à l'inverse est l'étude des procédés cryptographiques, qui dépendent d'un paramètre appelé clé [40],[41].

Dans ce chapitre on va parler de la cryptographie utilisée pour sécuriser ses objets connectes qui a la fois légère à cause des ressources limités.

III.1. Définition de la cryptologie

La cryptographie est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse. Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-d permettant de les rendre inintelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préférera le verbe chiffré. La cryptanalyse, à l'inverse, est l'étude des procédés cryptographiques dans le but de trouver des faiblesses, en particulier, de pouvoir décrypter des messages chiffrés. Le décryptement est l'action consistant à trouver le message en clair sans connaitre la clef de déchiffrement. La cryptologie, étymologiquement « la science du secret », ne peut être vraiment considérée comme une science que depuis peu de

temps. Cette science englobe la cryptographie (l'écriture secrète) et la cryptanalyse (l'analyse de cette dernière).

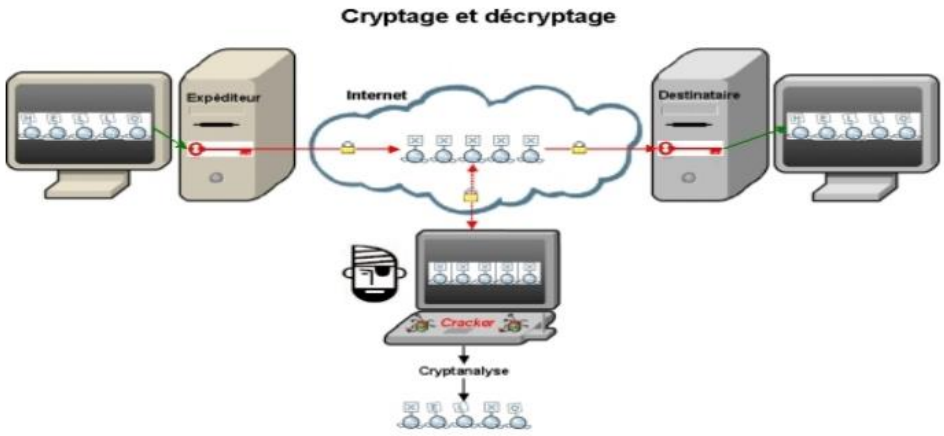


Figure III.1. Schéma de cryptage

III.2. Définition de la cryptographie

La cryptographie est l'art de chiffrer, coder les messages est devenue aujourd'hui une science à part entière. Au croisement des mathématiques, de l'informatique, et parfois même de la physique, elle permet ce dont les civilisations ont besoin depuis qu'elles existent : le maintien du secret. Pour éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses.

III.3. L'usage de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des

données mais aussi à garantir leur intégrité et leur authenticité.

- **La confidentialité :** consiste à rendre l'information intelligible à d'autres
- **L'intégrité :** vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication.
- **L'authentification :** consiste à assurer l'identité d'un utilisateur, c.-à-d. de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.
- **La non répudiation :** de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction [42].

III.4. Mécanisme de la cryptographie

Un algorithme de cryptographie ou un chiffrement est une fonction mathématique utilisée lors du processus de cryptage et de décryptage. Cet algorithme est associé à une clé (un mot, un nombre ou une phrase), afin de crypter une donnée. Avec des clés différentes, le résultat du cryptage variera également. La sécurité des données cryptées repose entièrement sur deux éléments : l'invulnérabilité de l'algorithme de cryptographie et la confidentialité de la clé [43].

Qu'entend-on par clé ?

On appelle clé une valeur utilisée dans un algorithme de cryptographie, afin de chiffrer une donnée. Il s'agit en fait d'un nombre complexe dont la taille se mesure en bits. On peut imaginer que la valeur correspondant à 1024 bits est absolument gigantesque. Voir aussi bits bytes. Plus la clé est grande, plus elle contribue à élever la sécurité à la solution. Toutefois, c'est la combinaison d'algorithme complexe et de clés importantes qui seront la garantie d'une solution bien sécurisée. Les clés doivent être stockées de manière sécurisée et de

manière à ce que seul leur propriétaire soit en mesure de les atteindre et de les utiliser [44].

III.5. Confidentialité et algorithmes de chiffrement

La confidentialité est le premier problème posé à la cryptographie. Il se résout par la notion de chiffrement. Il existe deux grandes familles d'algorithmes cryptographiques à base de clef : Les algorithmes à clef secrète ou algorithmes symétriques, et les algorithmes à clef publique ou algorithmes asymétriques.

- Chiffrement symétrique ou clef secrète : dans la cryptographie conventionnelle, les clefs de chiffrement et de déchiffrement sont identiques : c'est la clef secrète, qui doit être connue des tiers communiquant et d'eux seuls. Le procédé de chiffrement est dit symétrique.

Les algorithmes symétriques sont de deux types :

- Les algorithmes de chiffrement en continu, qui agissent sur le message en clair un bit à la fois
- Les algorithmes de chiffrement par bloc, qui opèrent sur le message en clair par groupes de bits appelés blocs.

Les principaux algorithmes à clé privée sont : Blowfish, DES/3DES, IDEA

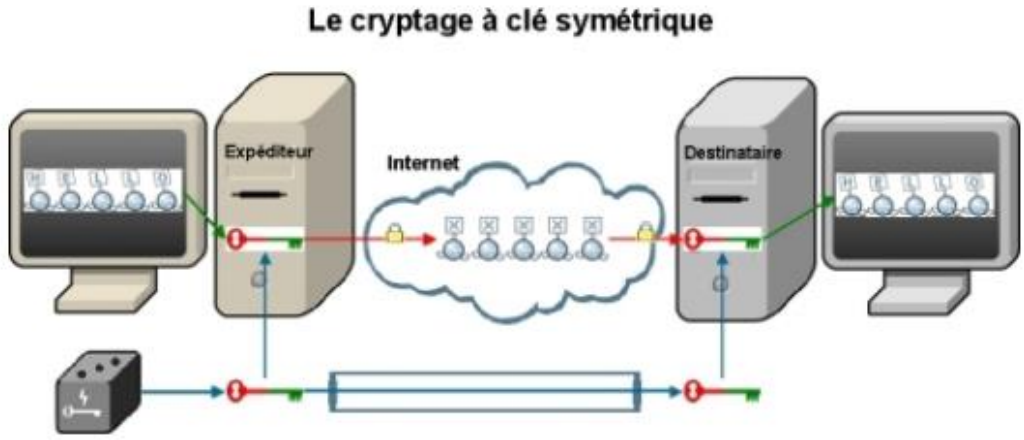


Figure III.2. Cryptage a clé symétrique

- Chiffrement asymétrique ou à clef public : avec les algorithmes asymétriques, les clefs de chiffrement et de déchiffrement sont distinctes et ne peuvent se déduire l'une de l'autre. On peut donc rendre l'une des deux publique tandis que l'autre reste privée. C'est pourquoi on parle de chiffrement à clef publique. Si la clef publique sert au chiffrement, tout le monde peut chiffrer un message, que seul le propriétaire de clef privé pourra déchiffrer. On assure ainsi la confidentialité. Certains algorithmes permettent d'utiliser la clef privée pour chiffrer. Dans ce cas, n'importe qui pourra déchiffrer, mais seul le possesseur de clef privée peut chiffrer [45].

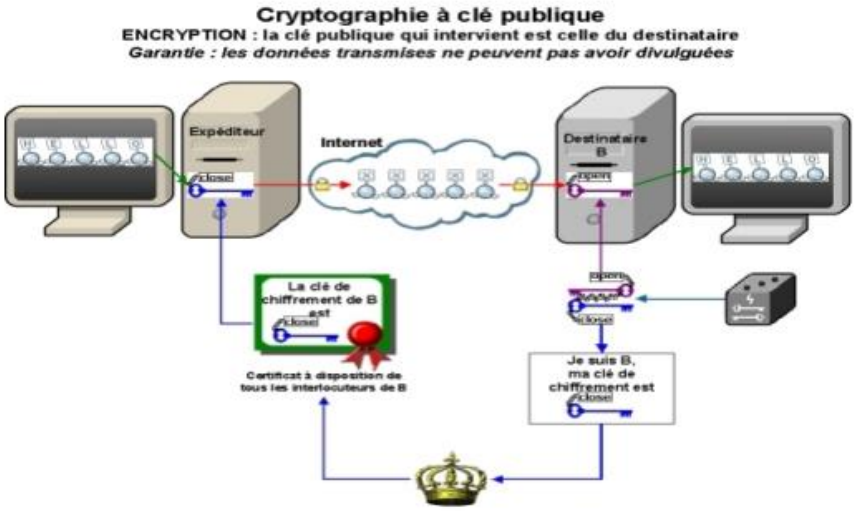


Figure III.3. Cryptage a clé publique

III.6. Les avantages et inconvénients de la cryptographie à clé publique comparé à la cryptographie à clé privée

Le premier avantage de la cryptographie à clé publique est d'améliorer la sécurité elle permet d'échanger des messages de manière sécurisée sans aucun dispositif de sécurité. L'expéditeur et le destinataire n'ont plus besoin de partager des clés secrètes via une voie de transmission sécurisée. Les

communications impliquent uniquement l'utilisation de clés publiques et plus aucune clé privée n'est transmise ou partagée. Avec un système à clé secrète, au contraire, il existe toujours le risque de voir la clé récupérée par une personne tierce quand elle est transmise d'un correspondant à l'autre. Toute personne interceptant la clé lors d'un transfert peut ensuite lire, modifier et falsifier toutes les informations cryptées ou authentifiées avec cette clé. De la norme de cryptage de données DES au code secret de Jules César, la distribution des clés reste le problème majeur du cryptage conventionnel. (Autrement dit, comment faire parvenir la clé à son destinataire sans qu'aucune personne ne l'intercepte ?) les moyens à déployer pour garantir la distribution sécurisée des clés correspondants sont très onéreux, ce qui constitue un inconvénient supplémentaire. Le cryptage à clé publique représente une révolution technologique qui offert à tout citoyen la possibilité d'utiliser une cryptographie robuste. En effet, la cryptographie conventionnelle était auparavant la seule méthode pour transmettre des informations secrètes. Les couts d'institutions disposants de moyens suffisants, telles que gouvernements et banque. Un autre avantage majeur des systèmes à clé publique est qu'ils permettent l'authentification des messages par signature électronique, ce qui peut aussi servir devant un juge, par exemple. L'inconvénient des systèmes à clé publique est leur vitesse contrairement aux méthodes à clé secrète qui sont plus rapide. Ils sont particulièrement adaptés à la transmission de grandes quantités de données. Mais les deux méthodes peuvent être combinées de manière à obtenir le meilleur de leurs systèmes. Pour le cryptage, la meilleure solution est d'utiliser un système à clé publique pour crypter une clé secrète qui sera alors utilisée pour crypter fichiers et message [46].

III.7. Description de systèmes cryptographiques classiques

III.7.1. Algorithme de substitution

Le chiffrement par substitution, consiste à remplacer dans un message une ou plusieurs entités (généralement des lettres) par une ou plusieurs autres entités.

- **Substitution monoalphabétiques :** Consiste à remplacer chaque lettre du message par une autre lettre de l'alphabet.
- **Substitution polyalphabétique :** consiste à utiliser une suite de chiffres monoalphabétiques réutilisée périodiquement
- **Substitution homophonique :** permet de faire correspondre à chaque lettre du message en clair un ensemble possible d'autres caractères
- **Substitution de polygrammes :** consiste à substituer un groupe de caractères (polygramme) dans le message par un autre groupe de caractères.

III.7.2. Le chiffrement par césar

Ce code est l'un des plus anciens, utilisé par Jules César. Le principe de codage repose sur l'ajout d'une valeur constante à l'ensemble des caractères du message, ou plus exactement à leur code ASCII. Il consiste en une substitution mono-alphabétique : chaque lettre est remplacée ("substitution") par une seule autre ("mono-alphabétique"), selon un certain décalage dans l'alphabet ou de façon arbitraire. D'après Suétone, César avait coutume d'utiliser un décalage de 3 lettres : A devient D, B devient E, C devient F, etc. Il écrivait donc son message normalement, puis remplaçait chaque lettre par celle qui lui correspondait. Lorsque l'ajout de la valeur donne une lettre dépassant la lettre Z, il suffit de continuer en partant de A, ce qui revient à effectuer un modulo 26.

Tableau III.1. Le principe de César

CLAIR	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-> décalage = 3																										
CODE	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

On appelle clé le caractère correspondant à la valeur que l'on ajoute au message pour effectuer le cryptage. Dans notre cas la clé est C, car c'est la 3éme lettre de l'alphabet. Ce système de cryptage est certes simple à mettre en œuvre, mais il a pour inconvénient d'être totalement symétrique, cela signifie qu'il suffit de faire une soustraction pour connaitre le message initial. Une méthode primaire peut consister à une bête soustraction des nombres 1 à 26 pour voir si l'un de ces nombres donne un message compréhensible. Une méthode plus évoluée consiste à calculer les fréquences d'apparition des lettres dans le message codé (cela est d'autant plus facile à faire que le message est long). Effectivement, selon la langue, certaines lettres reviennent plus couramment que d'autre (en français, par exemple, la lettre la plus utilisée est la lettre E), ainsi la lettre apparaissant le plus souvent dans un texte crypté par le chiffrage de César correspondra vraisemblablement à la lettre E, une simple soustraction donne alors la clé de cryptage [47], [48], [49].

III.7.3. Le chiffre de VIGENERE ou de BEAUFORT

Fonctionnement

Le chiffre de vigenere est un système de chiffrement, élaboré par Blaise de vigenere (1523- 1596), diplomate français du XVI° siècle.

Ce chiffrement introduit la notion de clé (elle se représente sous la forme d'un mot ou d'une phrase)

		Lettre en clair																									
		A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Cl	**A**	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Le

Tableau III.2. Table de vigenere

é	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	t
	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	t
U	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	r
t	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	e
i	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	
l	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	c
i	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	h
s	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	i
é	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	f
e	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	f
	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	r
	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	é
	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	e
	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	

Il consiste à remplacer une lettre par une autre qui n'est pas toujours la même. L'outil indispensable de ce chiffrement est la table de « vigenere »; table de 26 alphabets de substitution. Caractère de la clé K : nombre de décalage dans l'i_éme alphabet. Voici la table de vigenere. C'est un code très difficile à « casser » si on ne connait pas la clé, donc très sûr. Le codage/décodage est par contre un peu long… [43], [45], [48]. Pour coder ton message, pour chaque lettre du message en clair tu sélectionne la colonne correspondante et pour une lettre de la clé tu sélectionne la ligne adéquate, puis au croisement de la ligne et de la colonne on trouve la lettre de chiffrement. Si ton message est plus long que ta clé, réécris la première lettre de la clé. Par exemple si tu veux coder

« BONJOUR » avec la clé « SCOUT », ça donne « TQBDHMT », comme dans le tableau si dessous.

Message	clé	code
B	S	T
O	C	Q
N	O	B
J	U	D
O	T	H
U	S	M
R	C	T

Pour décoder il faut que tu connaisses la clé. Dans la colonne correspondant à la première lettre de la clé, trouve la première lettre du code. Tu peux donc retrouver la première du message au bout de la ligne. Poursuis ensuite avec les lettres suivantes. Là encore, si ton message codé est plus long que la clé, repars à la première lettre de la clé [48], [49].

III.8. Système cryptographiques moderne

III.8.1. Systèmes symétriques à clé secrètes

La cryptographie à algorithmes symétriques utilise la même clé pour les processus de codage et de décodage ; cette clé est le plus souvent appelée « secrète ». Le chiffrement consiste à appliquer une opération (algorithme) sur les données à chiffrer à l'aide de la clé privée, afin de les rendre inintelligibles. Ainsi, le moindre algorithme peut rendre le système quasiment inviolable (la sécurité absolue n'existant pas) [44]. Toutefois dans les années 40 Claude Shannon démontra qu'être totalement sûre, les systèmes à clefs privées doivent utiliser des clefs d'une longueur au moins égale à celle du message à chiffrer [41], [42], [47].

Principe de base

Un expéditeur et un destinataire souhaitant communiquer de manière sécurisée à l'aide du cryptage conventionnel doivent convenir d'une clé et ne pas la divulguer. Dans la majorité des systèmes de cryptages symétrique la clé de chiffrement et la clé de déchiffrement sont identiques.

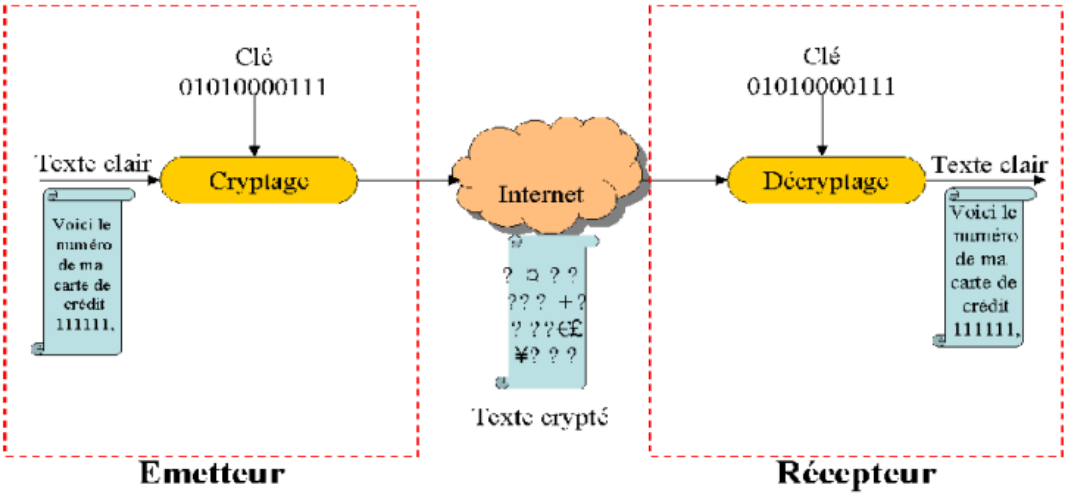


Figure III.4. Cryptographie symétrique

Quelques algorithmes de chiffrement symétriques très utilisés :

- Chiffre de Vernam (le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message, qu'elle ne soit utilisée qu'une seule fois à chiffrer et qu'elle soit totalement aléatoire)
- DES
- 3DES
- AES
- RC5

Génération du DES

Le Data Encription Standard (standard de chiffrement de données a été publié en 1977, et fut ainsi le premier algorithme cryptographie à petite clé secrète (56 bits) à avoir été rendu public. Le DES consiste en un réseau de Feistel de 16 tours : le message à chiffrer est découpé en blocs de 64 bits, chacun d'eux étant séparé en deux sous-blocs de 32 bits.

Les grandes lignes de l'algorithme sont les suivantes

Phase1 : préparation- diversification de la clé :

Le texte est découpé en blocs de 64 bits. On diversifie aussi la clé K, c'est-à-dire qu'on fabrique à partir de K 16 sous-clés K1,……, K16 à 48 bits. Les Ki sont composés de 48 bits de K, pris dans un certain ordre.

Phase2 : permutation initiale :

Pour chaque bloc de 64 bits x du texte, on calcule une permutation finie y=P(x). y est représenté sous la forme y=G0D0, G0 étant les 32 bits à gauche de y, D0 les 32 bits à droite.

Phase3 : Itération :

On applique 16 rondes d'une même fonction. A partir de Gi-1Di-1 (pour i de 1 à 16), on calcule GiDi en posant :

- Gi=Di-1
- Di=Gi-1 XOR f(Di-1,Ki)

XOR est le ou exclusif bit à bit, et f est une fonction de confusion, suite de substitution et de permutations.

Phase4 : permutation finale :

On applique à G16D16 l'inverse de la permutation initiale. $Z=P^{-1}(G16D16)$ est le bloc de 64 bits chiffré à partir de x.

Description du DES :

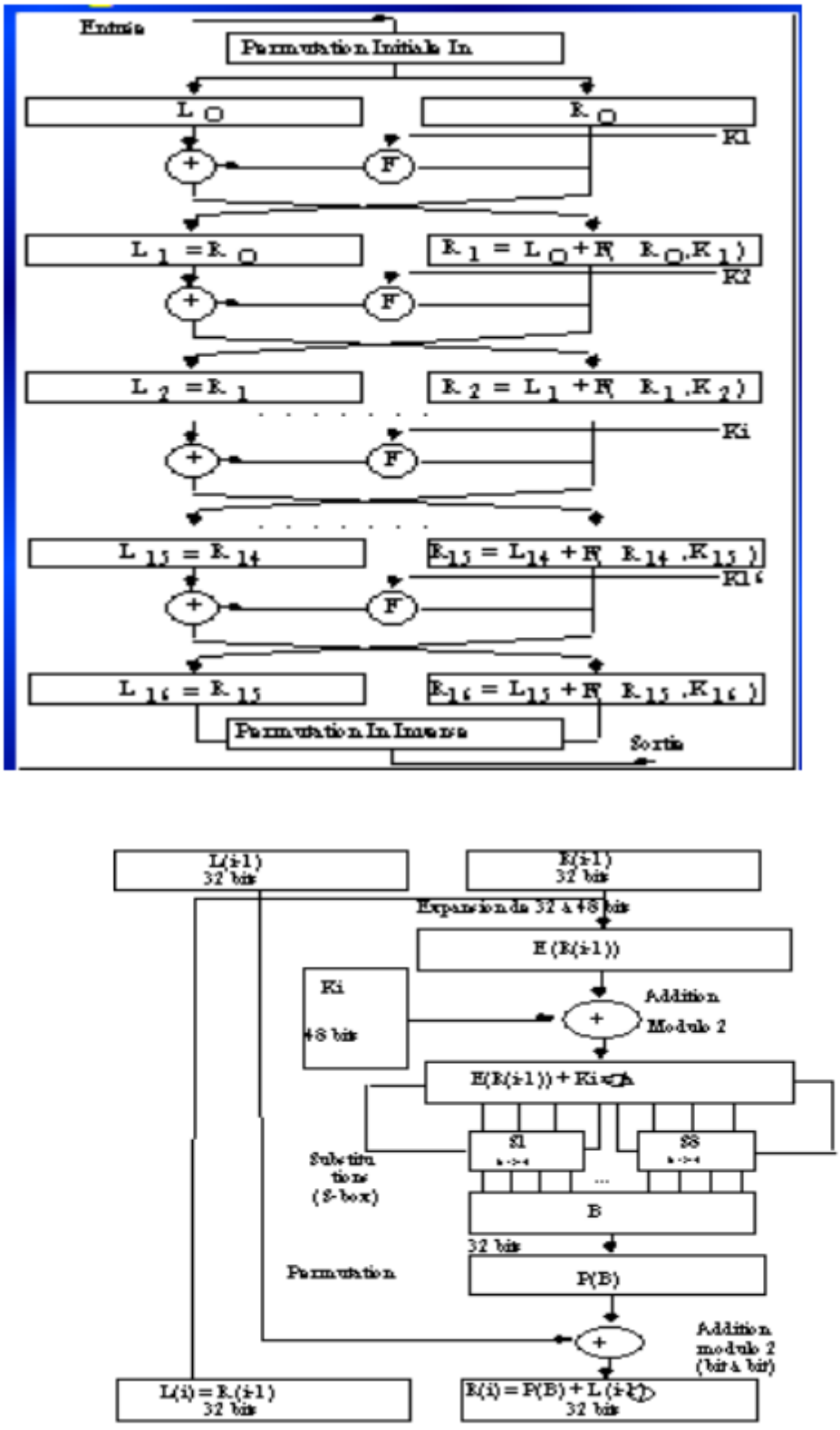


Figure III.5. Schémas générale du DES

III.8.2. Systèmes asymétriques à clé publique

Définition et fonctionnement

La cryptographie asymétrique à clé publique est apparue pour la première fois en 1976 avec la publication d'un ouvrage sur la cryptographie par Whitfield Diffie et Martin Hellman, c'est méthode de chiffrement qui s'oppose à la cryptographie symétrique.

Dans un tel crypto système, les clés existent en paires d'où l'appellation bi-clés :

- Une clé publique pour le chiffrement.
- Une clé secrète pour le déchiffrement.

L'utilisateur d'un crypto système asymétrique, choisit une clé aléatoire (la clé privé), à partir de cette clé et en appliquant la fonction à sens unique il calcule la clé publique qu'il diffuse au travers d'un canal non sécurisé. Lorsqu'une personne désire lui envoyer un message il lui suffit de chiffrer ce dernier à l'aide de la clé publique. Le destinataire sera en mesure de déchiffrer le message à l'aide de sa clé privé. Ce système est basé sur une fonction facile à calculer dans un sens (appelé fonction à trappe à sens unique) et mathématiquement très difficile à inverser sans la clé privée appelé trappe [24], [31], [47].

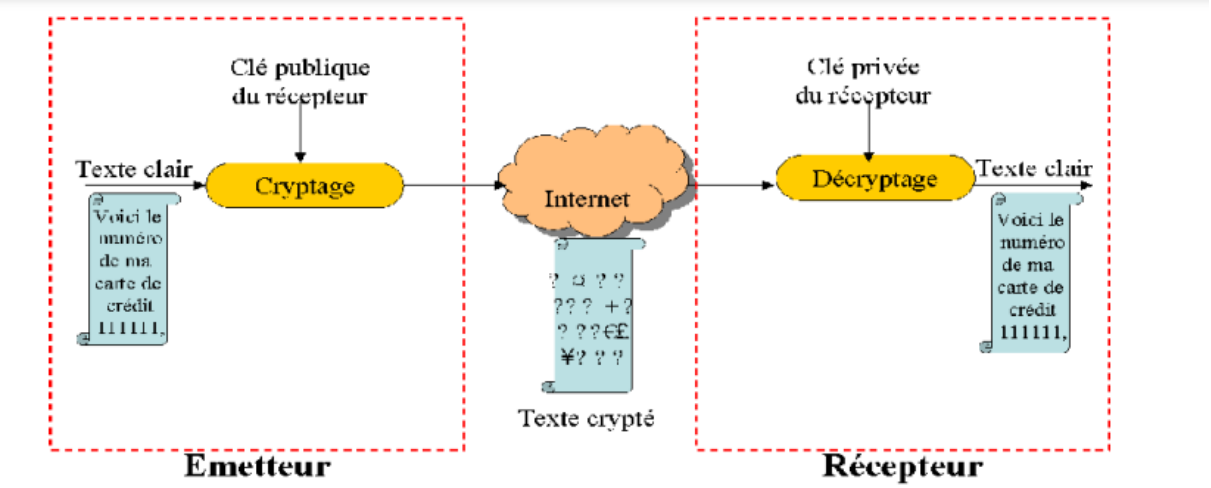


Figure III.6. Cryptographie asymétrique

Les principaux algorithmes asymétriques à clé publiques sont :

RSA (chiffrement et signature)

DSA (signature)

Diffie-Hellman (échange de clé) [35].

L'algorithme RSA

Le principe

Le premier système à clé publique solide à avoir été inventé, et le plus utilisé actuellement, est le système RSA. Publié en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman de l'Institution de technologie du Massachusetts, le RSA est fondé sur la difficulté de factoriser des grands nombres, et la fonction à sens unique utilisée est une fonction "puissance"

L'algorithme de chiffrement

Départ :

- Il est facile de fabriquer de grands nombres premiers p et q (+- 100 chiffres)
- Etant donné un nombre entier n = p*q, il est très difficile de retrouver les facteurs p et q

1) Création des clés

- La clé secrète : 2 grands nombres premiers p et q
- La clé publique : n = p*q ; un entier e premier avec (p-1) (q-1)

2) Chiffrement : le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante :

C = M^e mod n

3) Déchiffrement : il s'agit de calculer la fonction réciproque

M=C^d modn

tel que e.d = 1 mod [(p-1) (q-1)]

Exemple : chiffrer BONJOUR

1) Alice crée ses clés :

- La clé secrète : **p = 53, q = 97** (Note : en réalité, p et q devraient comporter plus de 100 chiffres !)
- La clé publique : **e = 7 (premier avec 52*96), n = 53*97 = 5141**

2) Alice diffuse sa clé publique (par exemple, dans un annuaire)

3) Bob ayant trouvé le couple (n, e), il sait qu'il doit l'utiliser pour chiffrer son message. Il va tout d'abord remplacer chaque lettre du mot BONJOUR par le nombre correspondant à sa position dans l'alphabet :

B = 2, O = 15, N = 14, J = 10, U = 21, R = 18

BONJOUR = 2 15 14 10 15 21 18

4) Ensuite, Bob découpe son message chiffré en blocs de même longueur représentant chacun un nombre plus petit que **n**. Cette opération est essentielle, car si on ne faisait pas des blocs assez longs (par exemple, si on laissait des blocs de 2 chiffres), on retomberait sur un simple chiffre de substitution que l'on pourrait attaquer par **l'analyse des fréquences**.

BONJOUR = 002 151 410 152 118

5) Bob chiffre chacun des blocs que l'on note B par la transformation C = Bemod n (où C est le bloc chiffré) :

C1 = 27mod 5141 = 128

C2 = 1517mod 5141 = 800

C3 = 4107mod 5141 = 3761

C4 = 1527mod 5141 = 660

C5 = 1187mod 5141 = 204

On obtient donc le message chiffré C : 128 800 3761 660 204

III.9. La Lightweight Cryptography

LE DPKI / WSN :

III.9.1. WIRELESS SENSOR NETWORK (WSN)

Un réseau de capteurs sans fil est un réseau ad hoc d'un grand nombre de nœuds, qui sont des micro-capteurs capables de recueillir et de transmettre des données d'une manière autonome. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement répartis dans une zone géographique, intitulée « champ de captage » correspondant au terrain concerné pour le phénomène capté [32].

Un réseau de capteurs sans fil (WSN) est composé d'un grand nombre de capteurs capables de surveiller leur environnement, de collecter de données et les transmettre à la station de base. Plusieurs WSN topologies ont été utilisées dans les applications de surveillance existantes : topologie distribuée, hiérarchique, centralisée ou décentralisée. Dans une topologie hiérarchique, le réseau est organisé en groupes. Chaque cluster a son propre Cluster Head (CH). Les nœuds d'un cluster communiquent avec le CH qui communique avec d'autres CH et la station de base (SB) pour relayer le données reçues et collectées. Un WSN se caractérise par l'énergie limitée, le stockage la capacité et la puissance de calcul des dispositifs capteurs souvent déployés pour des tâches sensibles, les communications doivent être sécurisées. Les mécanismes et architectures de sécurité dédiés sont à la fois sûrs et complexes. Les approches de chiffrement asymétrique sont une bonne solution pour WSN en raison de la taille nettement plus petite des clés à la même force cryptographique. Cependant, leur complexité de calcul est un obstacle à leur application. Par conséquent, plusieurs optimisations ont été faites, notamment sur la scalaire

multiplication d'un point, opération centrale et coûteuse en le cryptosystème à courbe elliptique (ECC) [33].

Dans ce contexte, nous proposons dans cet article un nouveau protocole de gestion de clés basé entièrement sur la cryptographie à clé publique (PKC), en utilisant le cryptosystème elliptique de courbe qui offre une certaine optimisation sur les calculs scalaires de multiplication et en les répartissant sur plusieurs nœuds du réseau. L'un des cryptosystèmes les plus utilisés est Elliptic Cryptographie de courbe **(ECC).** ECC utilise le même principe que RSA pour les opérations cryptographiques. Il est basé sur le problème du discret logarithme comme fonction à sens unique. Son principe est de trouver un entier k à partir de deux points de la courbe P et Q tel que Q = k · P. Le calcul dans un sens est facile mais en inversant ca supposé le problème NP-complet [34].

L'opération centrale dans ECC est la multiplication scalaire de un point (noté k · P où k est un scalaire et P est un point de la courbe elliptique). C'est une opération complexe qui nécessite des calculs intensifs.

Plusieurs optimisations ont été faites : la méthode NAF et w-NAF pour réduire le nombre d'opérations d'addition respectivement de $\frac{\log n}{2}$ à $\frac{\log n}{3}$ et $\frac{\log n}{w+1} + 2^{w-2}$ (où n est la taille de l'entier k en bits).
Chacune des méthodes tente de réduire le nombre d'additions et de doublements de points dans une multiplication scalaire. Une nouvelle méthode appelée Right-to-Left est appliquée sur l'algorithme standard Double-and-Add pour permettre des calculs parallèles (voir **Figure III.7**).

Require: k = $(k_{d-1},..., k_1, k_0)_2$, P ∈ E($F_p$)
Ensure: Q = k · P
  R_0 = O
  R_1 = P
  for $i \leftarrow 1$ to d-1 **do**
    if k_i = 1 **then**
      $R_0 \leftarrow R_0 + R_1$
    end if
    $R_1 \leftarrow 2 \times R_1$
  end for
  return R_0;

Figure III.7. Algorithm 1 Right-to-left

III.9.2. DPKI POUR WSN

dPKI distribued Public Key Infrastructure : est un protocole de sécurité asymétrique et distribué. Les multiplications d'un point sont réparties et réalisées par plusieurs nœuds du réseau. Les doublages de points sont effectués par les nœuds maîtres, appelés CHM, et les ajouts de points par les nœuds ordinaires. Afin d'éviter le problème d'inversion d'un point elliptique, opération extrêmement coûteuse, nous avons choisi les coordonnées jacobiennes où un point P ∈ E(Fp) est représenté par les trois coordonnées X, Y, Z qui correspondent au point affine $\left(\frac{X}{Z^2}, \frac{Y}{Z^3}\right)$ [45].

L'équation de Weierstrassest dans la forme jacobienne devient $Y^2Z = X^3 + aXZ^2 + bZ^3$. Chaque nœud maître CHM calcule un point P de l'elliptique courbe E(Fp), une clé privée k, tous les doublements du point P (noté D[P]), le point Q = k · P et tous les doublements de le point Q (noté D[Q]). Toutes ces données pré-calculées seront être envoyé aux nœuds ordinaires.

Chaque nœud ordinaire calcule uniquement des ajouts de points utilisant les tableaux de points reçus D[P] et D[Q]. L'algorithme de multiplication scalaire dans dPKI, exécuté par CHM et nœuds ordinaires, est donné par les deux algorithmes 2 et 3 [45].

Require: P $\in$ E(F_p)
Ensure: D[P]
  D[0] = P
  for $i \leftarrow$ 1 to d-1 **do**
    D[i] $\leftarrow$ 2 $\times$ D[i-1]
  end for
  return D[P];

Figure III.8. Algorithm 2 Point doubling calculé par CHM

Require: k = (k_{d-1},..., k_1, k_0)$_2$, D[P]
Ensure: Q = k · P
  Q = O
  for $i \leftarrow$ 0 to d-1 **do**
    if k_i = 1 **then**
      Q = Q + D[i]
    end if
  end for
  return Q;

Figure III.9. Algorithm 3 Point addition calculé par Ordinary Node

III.10. Protocole de gestion des clés

Le protocole de gestion des clés que nous proposons est un protocole déterministe, distribué, asymétrique, coopératif et hiérarchique. Il garantit toutes les fonctions de base d'une gestion de clés tel que l'établissement de communications sécurisées, révocation, résilience, renouvellement, insertion de nouveaux nœuds, etc. Dans ce nouveau dispositif, quatre acteurs sont identifiés : la base station, on suppose qu'aucun attaquant ne peut la corrompre, le CHM [47].

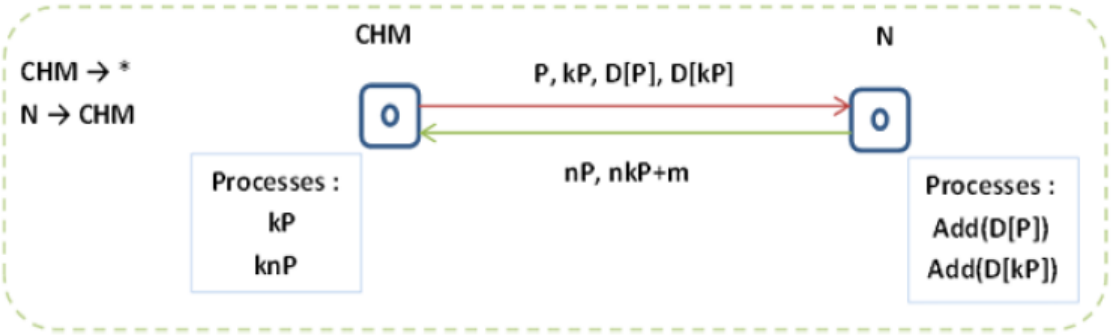


Figure III.10. Sécurité des messages envoyés entre le CHM et les nœuds ordinaires

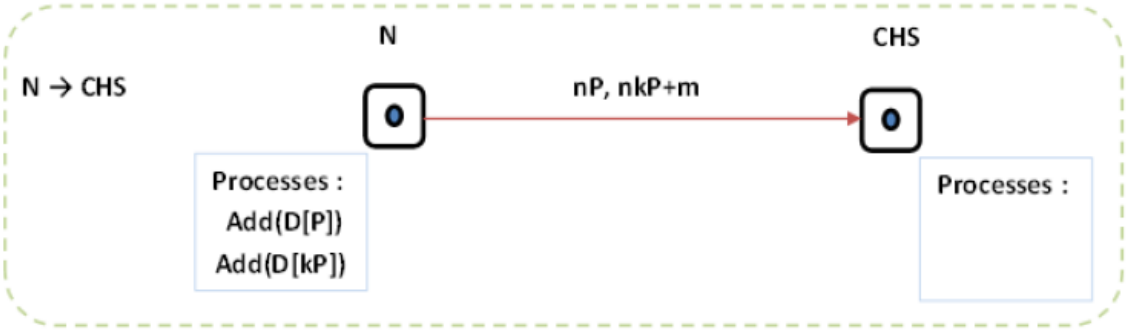


Figure III.11. Sécurité des messages envoyés entre les nœuds ordinaires, CHS et CHM.

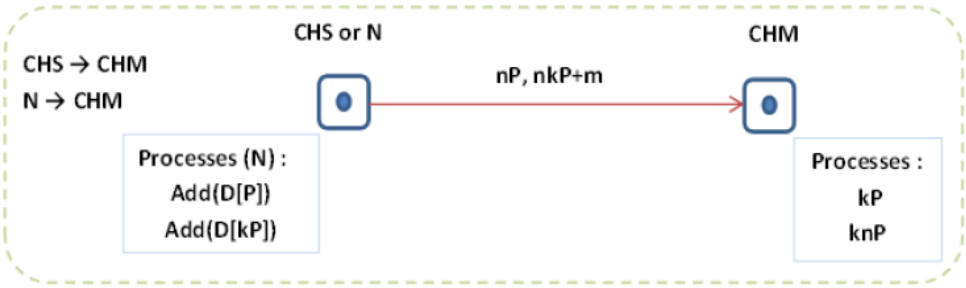


Figure III.12. Sécurité des messages envoyés entre les nœuds ordinaires et CHM.

Conclusion

Dans ce chapitre on a décrit quelques algorithmes de chiffrements les plus utilisés, mais bien-sûr il en existe beaucoup d'autres. Le choix de l'algorithme doit dépondre de l'application envisagée et donc des caractéristiques désirées et de l'espace mémoire disponible. Dans l'IOT il y a une spécification c'est que les ressources sont bien limitées donc il a fallu des cryptographies légères a fin s'approprier avec ces objets.

Chapitre IV :

Implémentation & Résultats

INTRDODUCTION

Nous présenterons dans ce chapitre les aspects techniques de l'implémentation nous allons faire des simulations via un logiciel appelé Arduino. Nous discuterons d' abord des choix d'outils logiciels utilisés, détaillerons ensuite quelques algorithmes pertinents dont on a besoin pour réaliser notre cryptographie légère pour des objets définis, enfin nous présenterons les résultats de nos développements tout en faisant une comparaison entre les types de chiffrement utilisés.

IV.1. Outils de simulation

Le choix des outils pour la simulation s'est fondé d' abord et avant tout sur le critère de simplicité. Nous avons testé plusieurs hypothèses avant d'opter pour un choix. Nous avons aussi considéré des facteurs secondaires comme par exemple la facilité de maintenance et la nécessité de disposer d'un système facile qui s'adapte a nos objets à ressources limités.

IV.1.1 Arduino

C'est un ensemble matériel et logiciel qui permet d'apprendre l'électronique (en s'amusant) tout en se familiarisant avec la programmation informatique. Arduino est en source libre ; vous pouvez donc télécharger le schéma d'origine et l'utiliser pour élaborer votre propre carte et la vendre sans payer des droits d'auteur [50].

IV.1.1.1. Matériel

Ce sont des cartes électroniques programmables (donc dotées d'un processeur et de mémoire) sur lesquelles nous pouvons brancher des capteurs de température, d'humidité, de vibration ou de lumière, une caméra, des boutons, des potentiomètres de réglage, des contacts électriques…

Il y a aussi des connecteurs pour brancher des LED, des moteurs, des relais, des afficheurs, un écran…

Une carte Arduino est un cerveau qui permet de rendre intelligent des

systèmes électroniques et d'animer des dispositifs mécaniques [50].

L'image ci-dessous montre une carte Arduino Uno qui est très utilisée pour débuter.

Figure IV.1. Une carte Arduino Uno avec ses connecteurs.

IV.1.1.2. Logiciel

Les créateurs de Arduino ont développé un logiciel pour que la programmation des cartes arduino soit visuelle, simple et complète à la fois. C'est ce que l'on appelle une IDE, qui signifie Integrated Development Environment ou Environnement de Développement « Intégré » en français (donc EDI).

L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino.

L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. Vous pouvez donc saisir votre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte arduino. A la date de rédaction de cette page, la version installée sur notre ordinateur de l'IDE Arduino est la 1.8.19. L'aspect est à peu près identique sur chaque plate-forme (Windows, Mac et Linux). L'image suivante montre l'écran initial qui apparaît au lancement de l'IDE [50].

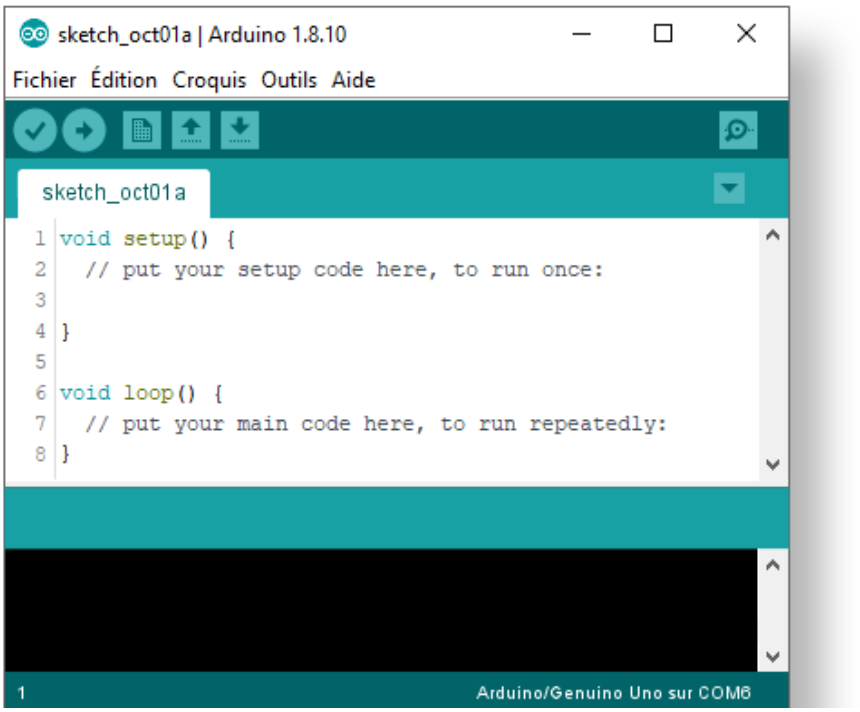


Figure IV.2. L'écran principal de l'IDE Arduino au démarrage

IV.1.1.3. Que peut-on faire avec une Arduino ?

En plus de la facilité de programmation rendue possible par l'IDE Arduino, l'autre grande caractéristique d'une Arduino est la capacité du micro-contrôleur sur lequel elle est basée. La condition est de garder à l'esprit quelques contraintes de base : mémoire, fréquence d'horloge, courants de sortie des périphériques et niveaux tensions.

Voici quelques applications possibles pour une Arduino :

- Mesure et détection
  - Station météorologique automatisée,
  - détecteur de foudre,
  - suivi du soleil pour orientation des panneaux solaires,
  - moniteur de radiation,
  - détecteur automatique de la faune,
  - système de sécurité domestique ou professionnel.

- Contrôle
  - Petits robots,
  - maquette de fusée ou d'avion,
  - drones multi-rotor,
  - CNC simple pour petites machines-outils.
- Automatisation
  - aquarium automatisé,
  - robot navette d'échantillon de laboratoire,
  - enceinte thermique de précision (couveuse, yaourtière, étuve, séchoir…),
  - système de test électronique automatisé.
- Art
  - contrôle d'éclairage et sonore dynamique,
  - structures cinématiques,

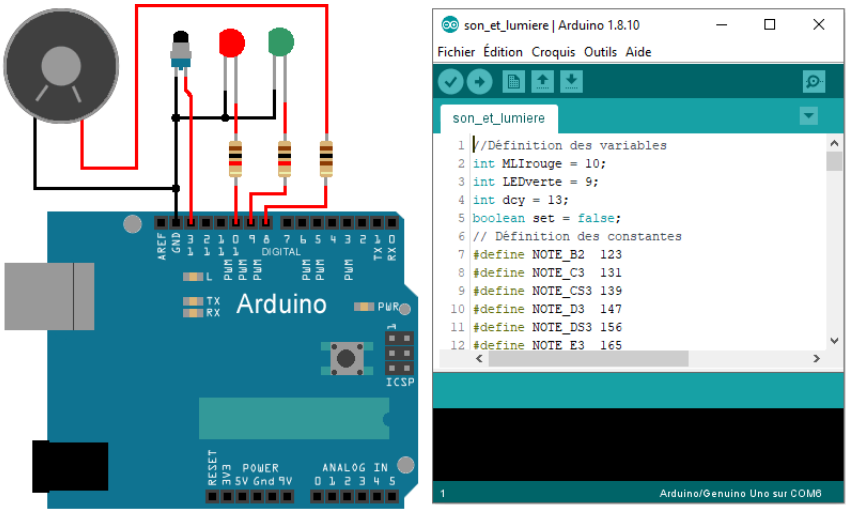


Figure IV.3. Un montage câblé avec une carte Arduino et son logiciel de programmation

.

IV.1.2. Tinkercad

On a utilisé aussi le logiciel en ligne tinkercad pour faire notre simulation sur Arduino Uno. Tinkercad est une collection d'outils logiciels en ligne et gratuite qui aide les utilisateurs du monde entier à inventer, à créer et à fabriquer. C'est l'outil idéal pour découvrir Autodesk, l'entreprise leader des logiciels de conception, d'ingénierie et de divertissement 3D

Grâce à Tinkercad on peut pour faire une programmation sur Arduino car on peut implémenter des circuits dedans, les images suivantes nous montre des captures d’écran sur logiciel Tinkercad sur web.

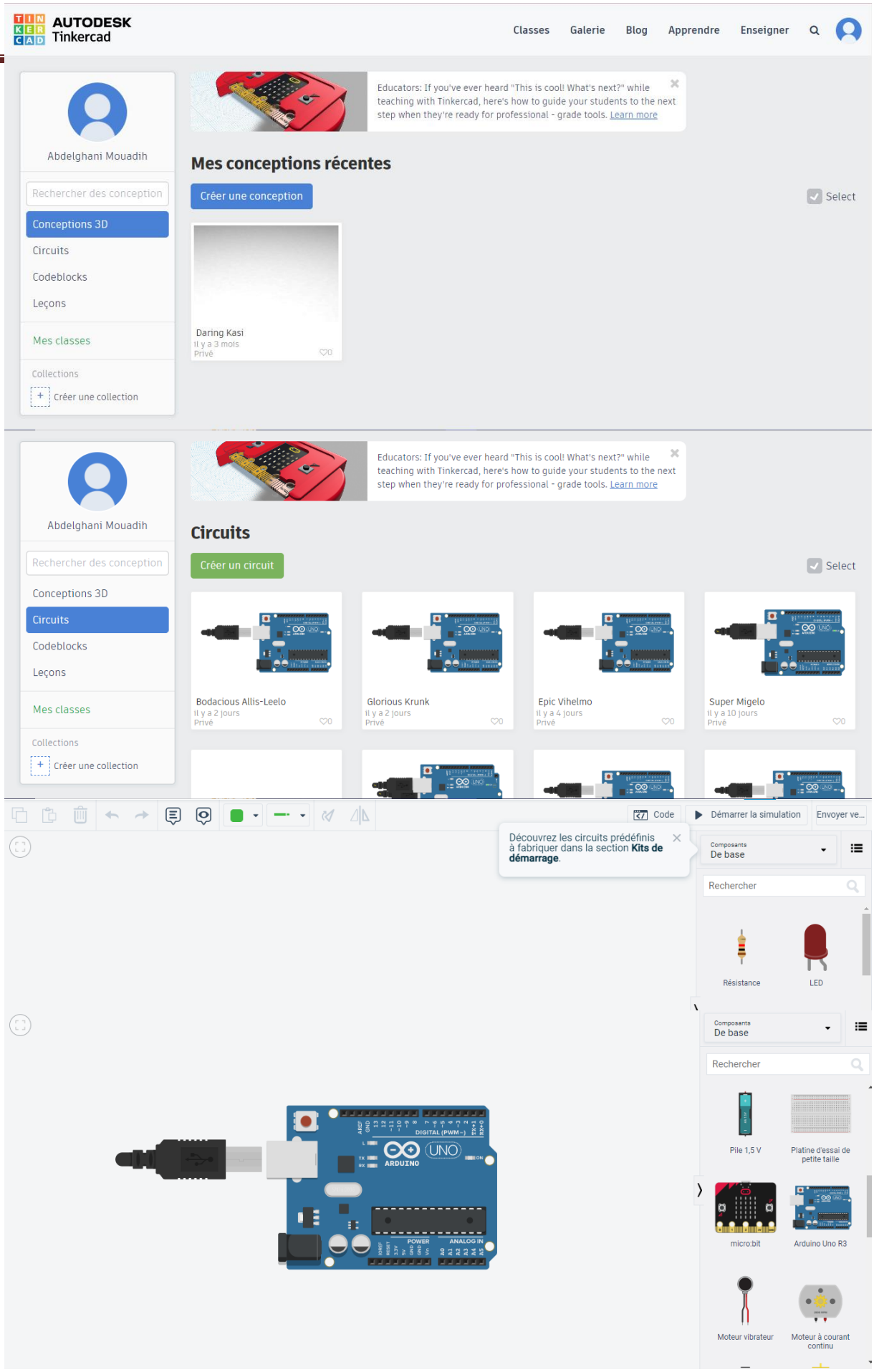


Figure IV.4. Interface Tinkercad

IV.2. Simulation du chiffrement par César Classique :

On a créé un circuit Arduino Uno R3, puis on a activé l'éditeur de code dans cette section, on a réalisé une simulation sur le loigiciel web Tinkercad en créant le circuit Arduino Uno R3 puis on exposera les résultats de simulations, ainsi que leurs discussions.

Tout d'abord on envisage le chiffrement par César classique ensuite on essaiera de l'améliorer en créant notre propre algorithme simplifié bien entendu car on est toujours sur l'idée de l'Internet des objets ou il nous faut des cryptographies simplifies et légères.

IV.2.1. Le principe du chiffrement Cesar classique

Le code César (ou chiffre de César) est un chiffrement par substitution mono alphabétique, où chaque lettre est remplacée par une autre lettre se situant un peu plus loin dans l'alphabet (donc décalée mais toujours identique pour un même message).

Cette méthode de cryptage est considérée comme le plus ancien des algorithmes de chiffrage par substitution, dans la mesure où Jules César l'aurait utilisé.

IV.2.2. Mécanisme :

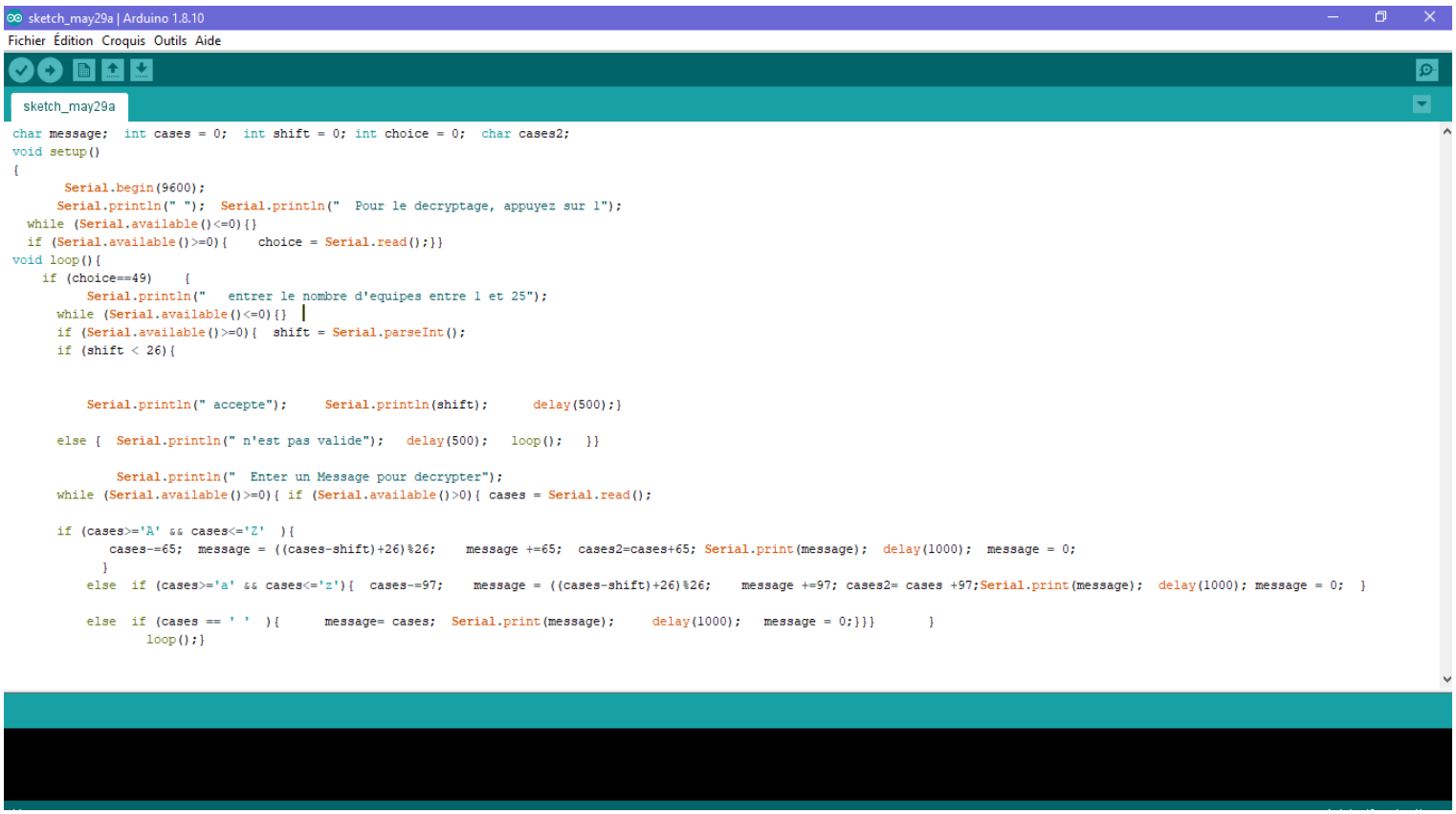
La technique est élémentaire : il suffit de remplacer chaque lettre du texte à chiffrer par la lettre qui se situe n places plus loin dans l'alphabet. Par exemple si n=3, on remplacera A par D, B par E, C par F etc.

Dans le cas spécifique du chiffrement de Jules César où la clé de cryptage est N (13ème lettre de l'alphabet), on appelle ce cryptage ROT13 (le nombre 13, la moitié de 26) a été choisi pour pouvoir crypter et décrypter facilement les messages. Dans le même ordre d'idée, le chiffre* Pigpen (utilisé autrefois par les francs-maçons), le Chiffre des Templiers ou le chiffre des Hommes Dansants remplacent chaque lettre par un symbole géométrique ou un dessin.

Le texte chiffré s'obtient en remplaçant chaque lettre du texte clair original par

une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet. Pour les dernières lettres (dans le cas d'un décalage à droite), on reprend au début. Par exemple avec un décalage de 3 vers la droite, A est remplacé par D, B devient E, et ainsi jusqu'à W qui devient Z, puis X devient A etc. Il s'agit d'une permutation circulaire de l'alphabet. La longueur du décalage, 3 dans l'exemple évoqué, constitue la clé du chiffrement qu'il suffit de transmettre au destinataire — s'il sait déjà qu'il s'agit d'un chiffrement de César — pour que celui-ci puisse déchiffrer le message. Dans le cas de l'alphabet latin, le chiffre de César n'a que 26 clés possibles [51].

- **Cryptage**



```
char message;  int cases = 0;  int shift = 0; int choice = 0;  char cases2;
void setup()
{
       Serial.begin(9600);
      Serial.println(" ");  Serial.println("  Pour le decryptage, appuyez sur 1");
  while (Serial.available()<=0){}
  if (Serial.available()>=0){    choice = Serial.read();}}
void loop(){
    if (choice==49)    {
         Serial.println("   entrer le nombre d'equipes entre 1 et 25");
      while (Serial.available()<=0){}
      if (Serial.available()>=0){  shift = Serial.parseInt();
      if (shift < 26){


          Serial.println(" accepte");     Serial.println(shift);      delay(500);}

      else {  Serial.println(" n'est pas valide");   delay(500);   loop();   }}

             Serial.println("  Enter un Message pour decrypter");
      while (Serial.available()>=0){ if (Serial.available()>0){ cases = Serial.read();

      if (cases>='A' && cases<='Z'  ){
            cases-=65;  message = ((cases-shift)+26)%26;    message +=65;  cases2=cases+65; Serial.print(message);  delay(1000);  message = 0;
          }
          else  if (cases>='a' && cases<='z'){  cases-=97;    message = ((cases-shift)+26)%26;    message +=97; cases2= cases +97;Serial.print(message);

          else  if (cases == ' '  ){      message= cases;  Serial.print(message);     delay(1000);   message = 0;}}}       }
                 loop();}
```

Figure IV.5. Le code source en langage Arduino du chiffrement Cesar

- **Decryptage**

```
char message;  int cases = 0;  int shift = 0; int choice = 0;  char cases2;
void setup()
{
       Serial.begin(9600);
      Serial.println(" ");  Serial.println("  Pour le decryptage, appuyez sur 1");
  while (Serial.available()<=0){}
  if (Serial.available()>=0){    choice = Serial.read();}}
void loop(){
    if (choice==49)    {
         Serial.println("   entrer le nombre d'equipes entre 1 et 25");
      while (Serial.available()<=0){}
      if (Serial.available()>=0){  shift = Serial.parseInt();
      if (shift < 26){


          Serial.println(" accepte");     Serial.println(shift);      delay(500);}

      else {  Serial.println(" n'est pas valide");   delay(500);   loop();   }}

             Serial.println("  Enter un Message pour decrypter");
      while (Serial.available()>=0){ if (Serial.available()>0){ cases = Serial.read();

      if (cases>='A' && cases<='Z'  ){
            cases-=65;  message = ((cases-shift)+26)%26;    message +=65;  cases2=cases+65; Serial.print(message);  delay(1000);  message = 0;
          }
          else  if (cases>='a' && cases<='z'){  cases-=97;    message = ((cases-shift)+26)%26;    message +=97; cases2= cases +97;Serial.print(message);  delay(1000); message = 0;  }

          else  if (cases == ' '  ){      message= cases;  Serial.print(message);     delay(1000);   message = 0;}}}       }
                 loop();}
```

Figure IV.5. Le code source en langage Arduino du déchiffrement Cesar

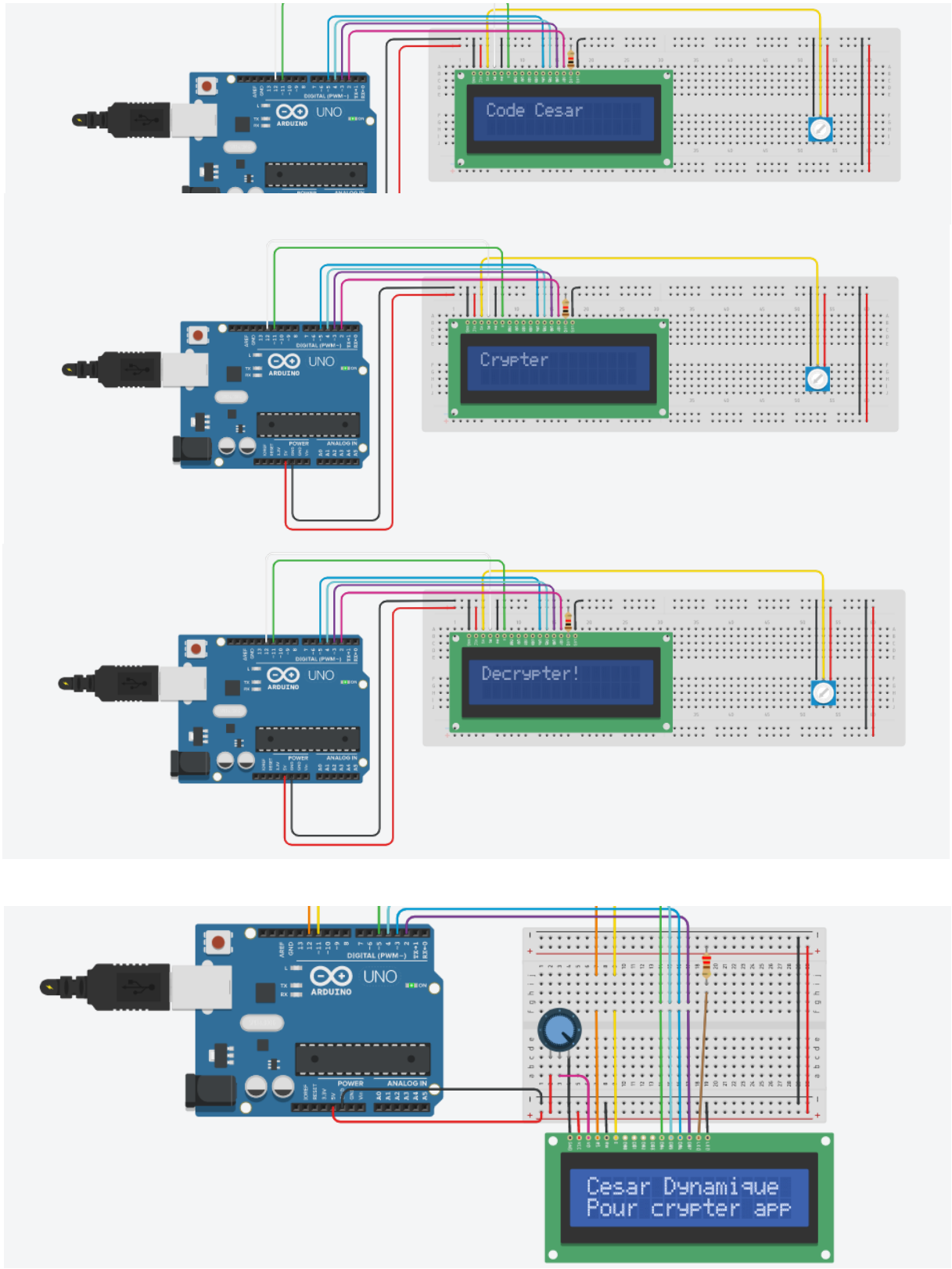


Figure IV.6. Exemple d'exécution avec un afficheur LCD

IV.2.3. Application de Chiffrement Cesar :

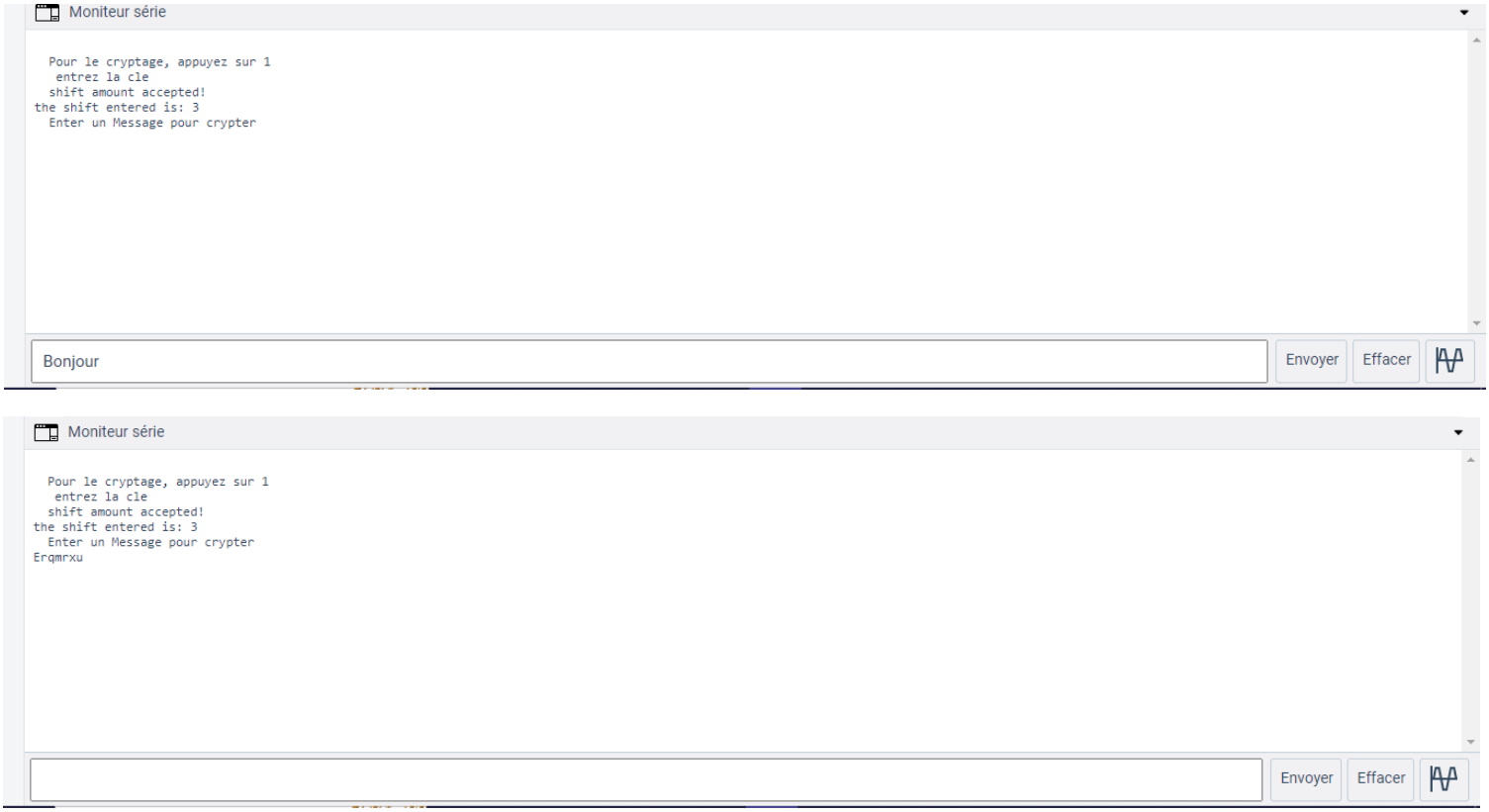


Figure IV.7. Exemple d'exécution d'un message ''Bonjour''

1- On commence par crypter sur trois exemples différents (avec un décalage

- **Paragraphe 1 :** '' *La guerre d'Algérie voit s'affronter l'armée française et les nationalistes algériens''*
- **Paragraphe 2 :** *''L'Algérie revendique depuis longtemps son indépendance alors qu'un large mouvement de décolonisation a lieu après la Seconde Guerre mondiale dans le monde entier.''*
- **Paragraphe 3 :** *'' Pour les Algériens, la lutte armée sert à exprimer une désillusion réelle à l'égard des promesses françaises. L'Algérie revendique depuis longtemps son indépendance alors qu'un large mouvement de décolonisation a lieu après la Seconde Guerre mondiale dans le monde entier*

2- Ça nous donne comme résultats :

- **Paragraphe 1 :** *''Nc iwgttg fCnitkg xqkv uchhtqpvgt nctog htcpckug gv ngu pc''*
- **Paragraphe 2 :** *''Nc iwgttg f'Cnitkg xqkv uchhtqpvgt nctog htcpckug gv ngu pcvkqpcnkuvgu cnitkgpu''*
- **Paragraphe 3 :''** *N'Cnitkg tgxgpfkswg fgrwku nqpivgoru uqp kpfrgpfcpeg cnqtu sw'wp nctig oqwxgogpv fg feqnqpkucvkqp c nkgw crtu nc Ugeqpfg Iwgttg oqpfkcng fcpu ng oqpfg feqnqpkucvkqp c nkgw crtu nc Ugeqpfg Iwgttg oqpfkcng fcpu ng oqpfg gpvkgt.''*

IV.2.4. Calcul de temps

- On va calculer le temps de simulation pour chaque cryptage et décryptage avec la fonction suivante

micros()

```
unsigned long time;

void setup() {
  Serial.begin(9600);
}
void loop() {
  Serial.print("Time: ");
  time = micros();

  Serial.println(time); //prints time since program started
   delay(1000);          // wait a second so as not to send massive amounts of data
} delay(1000);          // wait a second so as not to send massive amounts of data
    }
```

Figure IV.8. la fonction micros() qui calcule le temps

Cette fonction Renvoie le nombre de microsecondes depuis que la carte Arduino a commencé à exécuter le programme en cours. Ce nombre va déborder (revenir à zéro), après environ 70 minutes**.**

```
 Pour lancer le programme, appuyez sur A
   entrer la clÃ© entre 1 et 25
  accepter
la cle : 2
 pour crypter écrire 1, pour décrypter écrire 2
 M  accepter
le M:1
  Enter un Message pour crypter ou décrypter
t   Time: 1322172
g   Time: 1322172
f   Time: 1322172
C   Time: 1322172
n   Time: 1322172
i   Time: 1322172
 Pour lancer le programme, appuyez sur A
   entrer la clÃ© entre 1 et 25
  accepter
la clé : 2
 pour crypter écrire 1, pour decrypter ecrir 2
 M  accepter
le M:2
  Enter un Message pour crypter ou décrypter
e  Time: 1015988
s  Time: 1015988
c  Time: 1015988
p  Time: 1015988
p  Time: 1015988
c  Time: 1015988 … jusqu'au a la fin du message clair
```

Figure IV.9. Calcul de temps de cryptage et decryptage

Le tableau suivant montrent les 3 temps en microseconde

Tableau IV.1. Tableau qui calcule le temps de cryptage et décryptage

	Temps de Cryptage	**Temps de décryptage**
Paragraphe 1	30,727,500 µs	46,251,560 µs
Paragraphe 2	58,201,500 µs	87,605,896 µs
Paragraphe 3	97,966,500 µs	147,460,856 µs

IV.2.3. Inconvénients :

Ce système de cryptage est très simple à mettre en œuvre, cependant étant totalement symétrique, il suffit de faire une soustraction pour connaître le message initial. Une méthode primaire est d'essayer les 26 combinaisons possibles et voir si l'on peut obtenir un message compréhensible. Une méthode

plus évoluée consiste à calculer les fréquences d'apparition des lettres dans le message codé (ce qui est beaucoup plus facile lorsque le message est long).

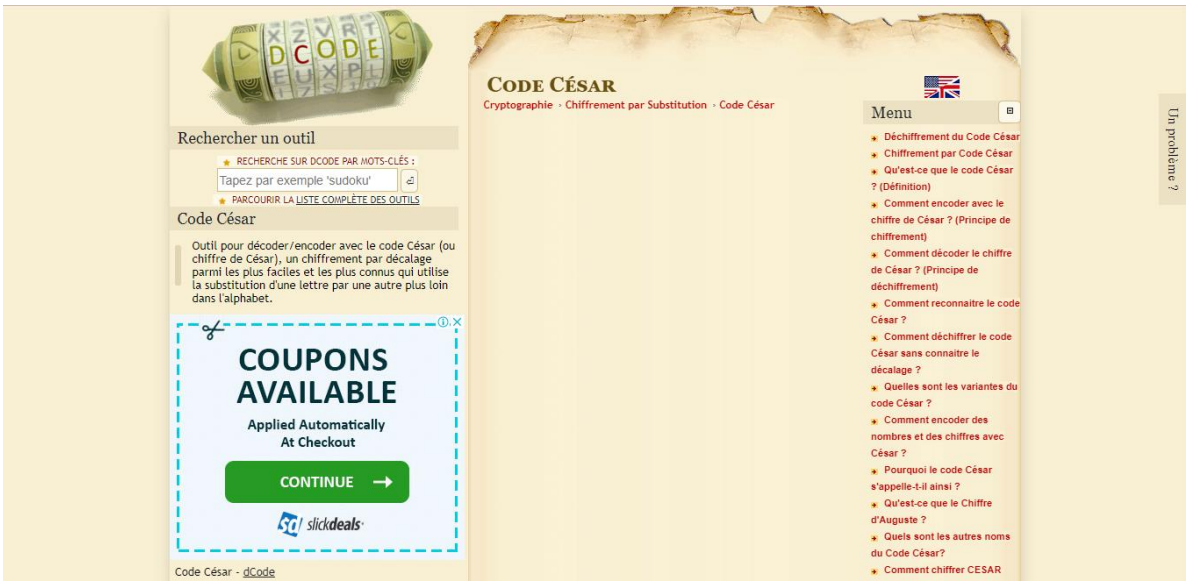
Effectivement, selon la langue, certaines lettres reviennent plus couramment que d'autres (en français, par exemple, la lettre la plus utilisée est la lettre E), ainsi la lettre apparaissant le plus souvent dans un texte crypté par le chiffrage de César correspondra vraisemblablement à la lettre E, une simple soustraction donne alors la clé de cryptage

IV.2.4. La Cassabilité d'un code Cesar :

Grace à la technique Force Brut la cassabilité d'un message chiffré par César devient banale

FORCE BRUT : Une attaque par **force brute** (bruteforce attack) consiste à tester, l'une après l'autre, chaque combinaison possible d'un mot de passe ou d'une clé pour un identifiant donné afin se connecter au service ciblé. Il s'agit d'une méthode ancienne et répandue chez les pirates [52].

On a testé la cassabilité d'un code chiffée par Cesar et on a donc trouvé un site web souvent utilisé par les tentateurs de décoder un message crypté [53].



On a essayé avec le premier cas celui de ''*Ils reconnaissent l'indépendance de l'Algérie''* avec le message chiffré *''Knu tgeqppckuugpv nkpfrgpfcpeg fg nCnitkg''* dans ce site et il l'a déchiffré en peu de temps.

Cela prouve la fébrilité de ce type de cryptage. C'est pour ça on a voulu proposer un concept inédit qui peut nous assurer et renforcer notre cryptographie tout en gardant le même principe de césar mais en améliorant son attitude de cryptanalyse.

Figure IV.10. Test de cassabilité César

IV.3. Simulation du chiffrement par César (amélioré) dynamique :

Cet algorithme de chiffrement nouveau est basé du classique celui du chiffrement de césar basé sur le décalage mais cette fois ci avec une clé dynamique qui change a chaque fois

Le but de faire cet algorithme c'est d'essayer d'améliorer le chiffrement classique mais avec de nouvelles méthodes

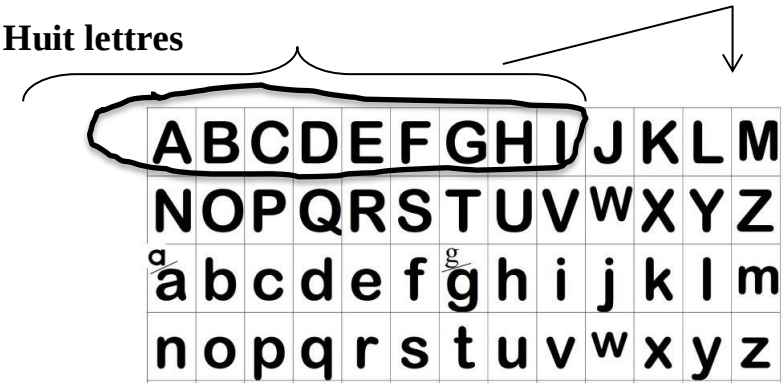
IV.3.1. Principe : on a fait une permutation entre les lettres dans l'ordre classique de l'alphabet françaises qui est constitué donc de 26 lettres cette permutation change dans 3cas

Tableau IV.2. Le tableau des 26 lettres

A	B	C	D	E	F	G	H	I	J	K	L	M
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	b	c	d	e	f	g	h	i	j	k	l	m
n	o	p	q	r	s	t	u	v	w	x	y	z

1er cas : Cryptage et décryptage par bloc 8 (permutation par 8 lettres)

Dans ce cas k=8 (la permutation est par 8 lettre) on a donc crée notre propre ordre alphabétique tout en changeant les positions des lettres par groupe de huit ensuite on donnera la main à l'utilisateur qui proposera sa propre clé de chiffrement. Et dans le cas inverse celui du décryptage, on enlevé la clé (soustraction) et puis on réorganise notre ordre des lettres du message chiffré.

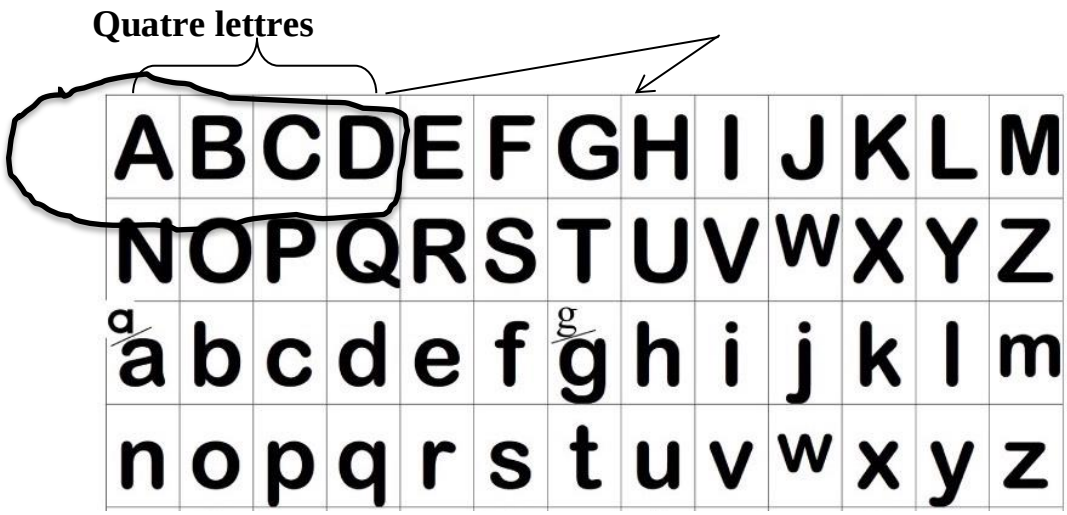


0	00	^@	32	20		64	40	@	96	60	`
1	01	^A	33	21	!	65	41	A	97	61	a
2	02	^B	34	22	"	66	42	B	98	62	b
3	03	^C	35	23	#	67	43	C	99	63	c
4	04	^D	36	24	$	68	44	D	100	64	d
5	05	^E	37	25	%	69	45	E	101	65	e
6	06	^F	38	26	&	70	46	F	102	66	f
7	07	^G	39	27	'	71	47	G	103	67	g
8	08	^H	40	28	(	72	48	H	104	68	h
9	09	^I	41	29	)	73	49	I	105	69	i
10	0A	^J	42	2A	*	74	4A	J	106	6A	j
11	0B	^K	43	2B	+	75	4B	K	107	6B	k
12	0C	^L	44	2C	,	76	4C	L	108	6C	l
13	0D	^M	45	2D	-	77	4D	M	109	6D	m
14	0E	^N	46	2E	.	78	4E	N	110	6E	n
15	0F	^O	47	2F	/	79	4F	O	111	6F	o
16	10	^P	48	30	0	80	50	P	112	70	p
17	11	^Q	49	31	1	81	51	Q	113	71	q
18	12	^R	50	32	2	82	52	R	114	72	r
19	13	^S	51	33	3	83	53	S	115	73	s
20	14	^T	52	34	4	84	54	T	116	74	t

Figure IV.11. Code ASCII

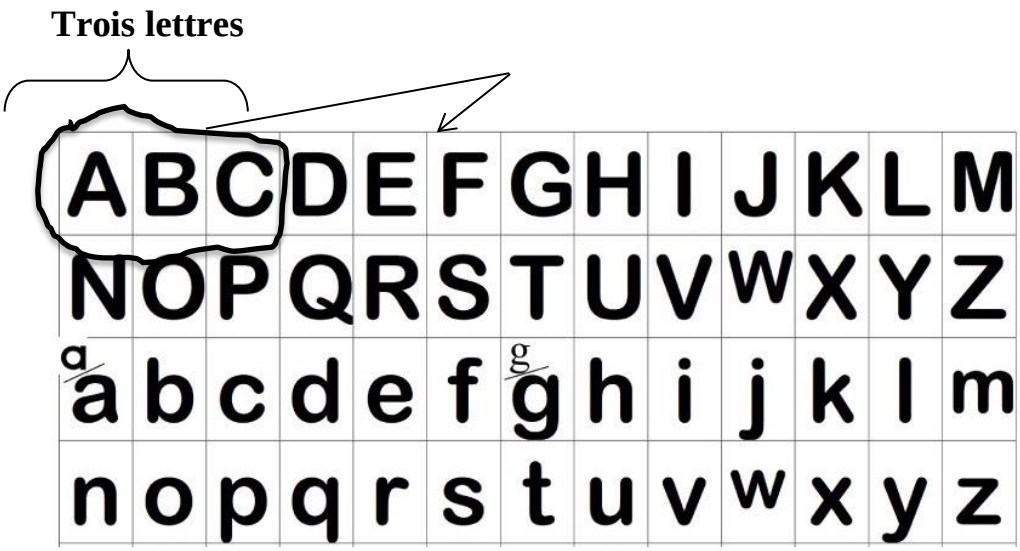
2er cas : Cryptage par bloc 4

Cryptage et décryptage par bloc 4 (permutation par 4 lettres)



3e cas : Cryptage par bloc 3

Cryptage et décryptage par bloc 3 (permutation par 3 lettres)



4eme cas : en utilisant les 4 cas
Déclaration des variables :
Chaine de caractère : message ;
Entier : k, clé, ltr ;
Initialisation : ltr = 0 ; clé = 0 ; K = 0 ;
Début
Entre une clé « décalage » ;
Teste le décalage entre 1 et 26 ;
Choisir la méthode A ou B ou C ;
Entre un message pour crypter ;
Si (k = A) alors : {
Chiffré avec Blok 3 ;
Diviser les lettre d'alphabet par group de 3 ;
Faire une permutation entre ces group ;
Chiffré le nouveau message avec le décalage «la clé » donnée ; }
Si non
Si (K = B) {
« Chiffré avec Blok 4 » ;
Diviser les lettre d'alphabet par group de 4 ;
Faire une permutation entre ces group ;
Chiffré le nouveau message avec le décalage «la clé » donnée ; }
Si non
Si (K = C) {
Chiffré avec Blok 8 ;
Diviser les lettre d'alphabet par group de 8 ;
Faire une permutation entre ces group ;
Chiffré le nouveau message avec le décalage «la clé » donnée ; }
Retourner (message crypter : « ») // affichage de résultat
Retourner (temps : « ») // Calculer et afficher le tempe
Fin.

Figure IV.12. : Algorithme du chiffrement Cesar dynamique

Message : BONJOUR, Décalge :2, Cas : Bloc4)

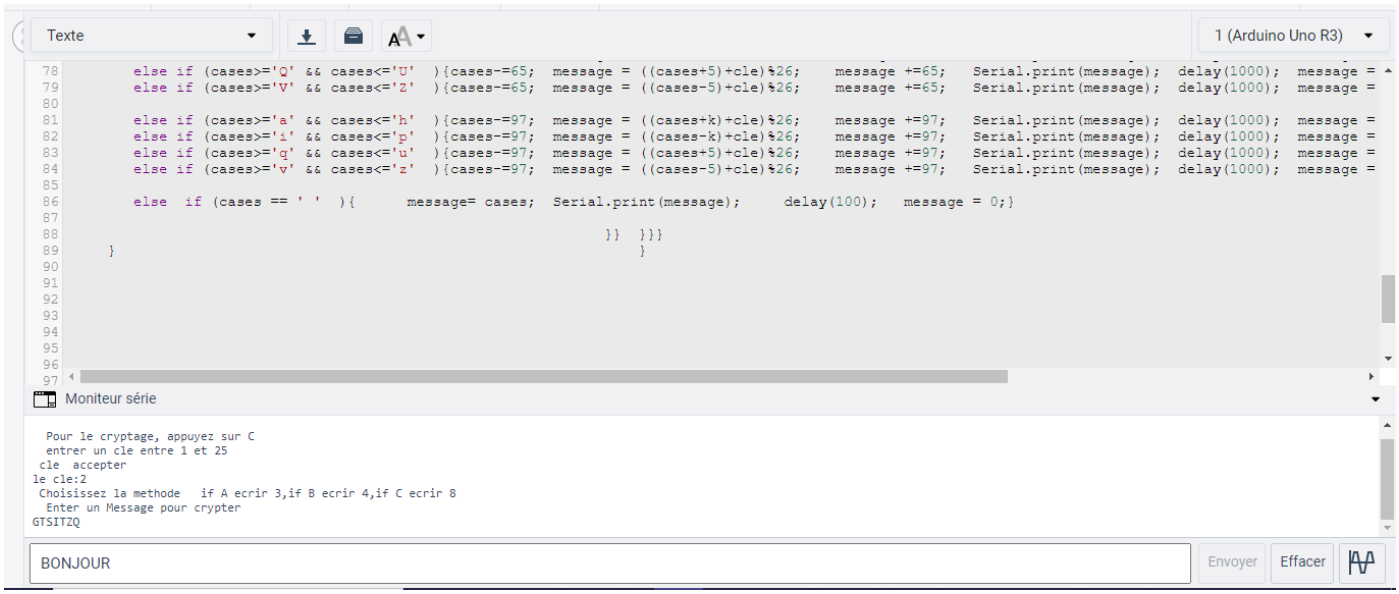


Figure IV.13. 4e cas : Cryptage et décryptage par les cas A, B, C

IV.3.2. Calcul de temps

On rechiffre les trois exemples précédents (avec un décalage de 2) et avec la même fonction **micros()**

```
Serial.print("Time: ");
time = micros();
```

Tableau IV.3. Tableau qui calcule le temps de cryptage et décryptage

	Temps de Cryptage	**Temps de décryptage**
Paragraphe 1	23,510,656 µs	23,938,360 µs
Paragraphe 2	53,312,896 µs	54,282,760 µs
Paragraphe 3	89,737,856 µs	91,370,360 µs

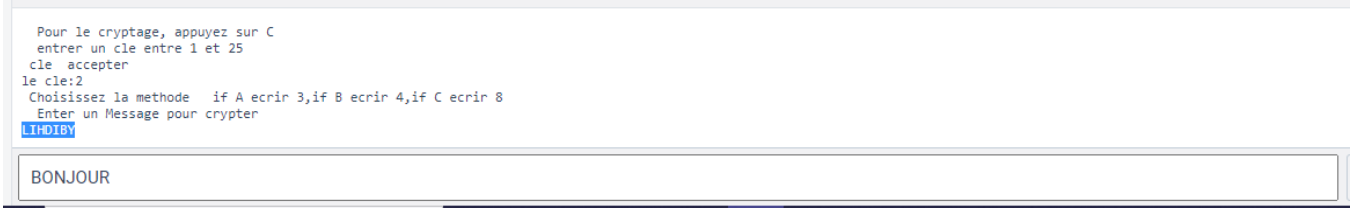
```
o   Time: 779332
s   Time: 779332
i   Time: 779332
c   Time: 779332
 Cryptage blok 8, appuyez sur C
  Entrer un clé entre 1 et 25
 clé  acceptée
la cle:2
 Pour crypter écrire 1, pour décrypter écrire 2
 M  accepter
le M:2
  Enter un Message pour crypter
B   Time: 922448
j   Time: 922448
b   Time: 922448
```

Figure IV.14. Calcul de temps de cryptage et décryptage

IV.3.2. La Cassabilité d'un code Cesar Dynamique :

Avec le même site précédent ‘’*https://www.dcode.fr/chiffre-cesar’’* on a testé la cassabilité de notre algorithme que nous avons proposé et on a su que notre systèmes de chiffrement est finalement fiable même difficile à décrypter tant que la méthode utilisée par les attaquants est force brut ou celle des essai des combinaisons et lettres répétés dans la langue française parcque on fait genre de doubler la façon de chiffrement qui ne se contente pas seulement de la clé mais aussi des blocs diviser et cette idée proposée a garanti la force de notre code.

Après qu'on a essayé de casser notre code mais n'a pas fonctionné.



→20 (←6)	RONJOHE
→19 (←7)	SPOKPIF
→7 (←19)	EBAWBUR
→23 (←3)	OLKGLEB
→3 (←23)	IFEAFYV
→16 (←10)	VSRNSLI
→14 (←12)	XUTPUNK
→21 (←5)	QNMINGD
→4 (←22)	HEDZEXU
→8 (←18)	DAZVATQ
→10 (←16)	BYXTYRO
→24 (←2)	NKJFKDA
→13 (←13)	YVUQVOL
→15 (←11)	WTSOTMJ

Figure IV.15. L'essai de cassabilité de notre message

Le message clair : BONJOUR
Clé : 2
Le message chiffré : LIHDIBY
Ce site a fait trop d'essais mais sans résultats vrais

IV.4. Remarques sur les résultats précédents :

IV.4.1. La complexité : on a remarqué que lorsque le programme est simple, la complexité est minime mais quand on a amélioré notre algorithme de chiffrement ''César Dynamique'' là on a bien supervisé que la complexité augmente

IV.4.2. Le temps de simulation : autre remarque bien évidente c'est celle du temps ça veut dire quand on crypte on prend moins de temps par rapport au décryptage

Les temps de cryptage se diffèrent également entre les deux algorithmes on observe dans les histogrammes suivants une comparaison entre les temps de cryptage et de décryptage sur un simulateur Tinkercad . Ce temps va vraisemblablement changer en temps réel et ça due a la vitesse réseau et la capacité mémoire et processeur sur un système puissant par rapport a la carte Arduino.

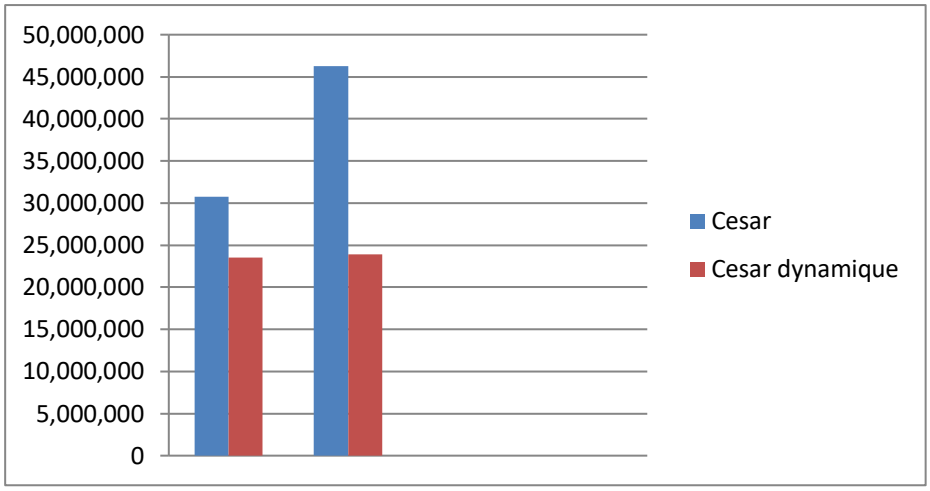


Figure IV.16. Temps de cryptage & décryptage César vs César dynamique du paragraphe 1

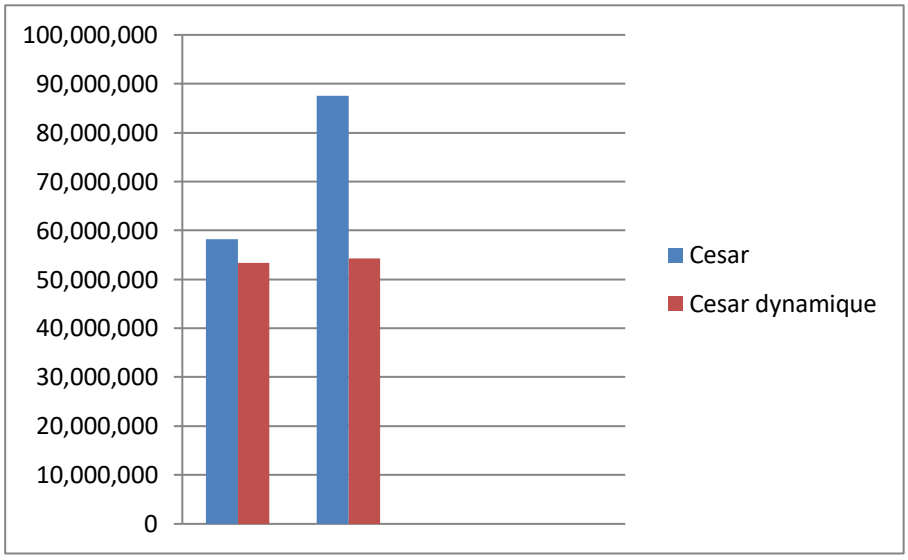


Figure IV.17. Temps de cryptage & décryptage César vs César dynamique du paragraphe 2

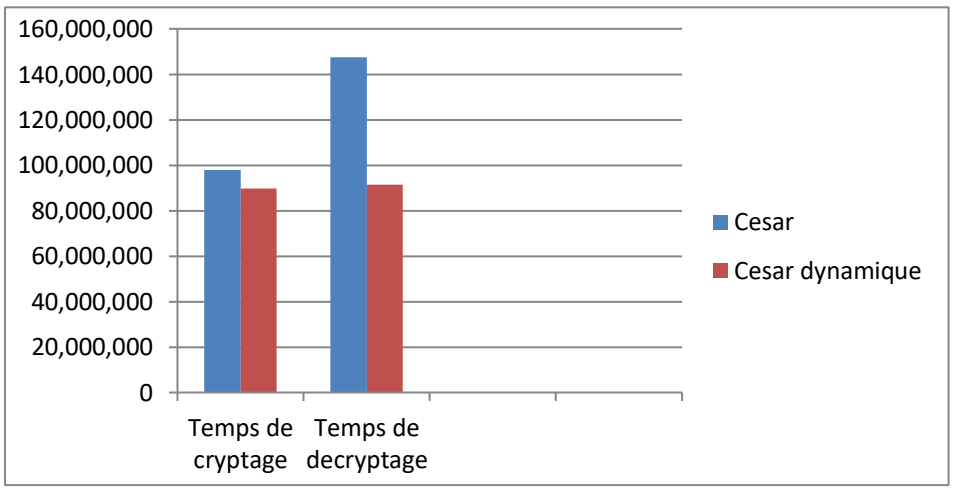


Figure IV.18. Temps de cryptage & décryptage César vs César dynamique du paragraphe 3

Les histogrammes précédents montrent une comparaison entre le temps décryptage et temps de cryptage pendant le chiffrement de césar classique en bleu et césar dynamique en rouge : on remarque que le temps pendant le chiffrement et déchiffrement de César Dynamique et bien inférieur à celui du précédent model de chiffrement et ça dû à la façon de cryptage de CESAR qui chiffre tout une chaine de caractère en effectuant le décalage tandis que pour CESAR Dynamique on crypte par block de lettres ; ce qui rend la tâche facile tout ben prenant moins de temps .

IV.4. Simulation du chiffrement par RSA :

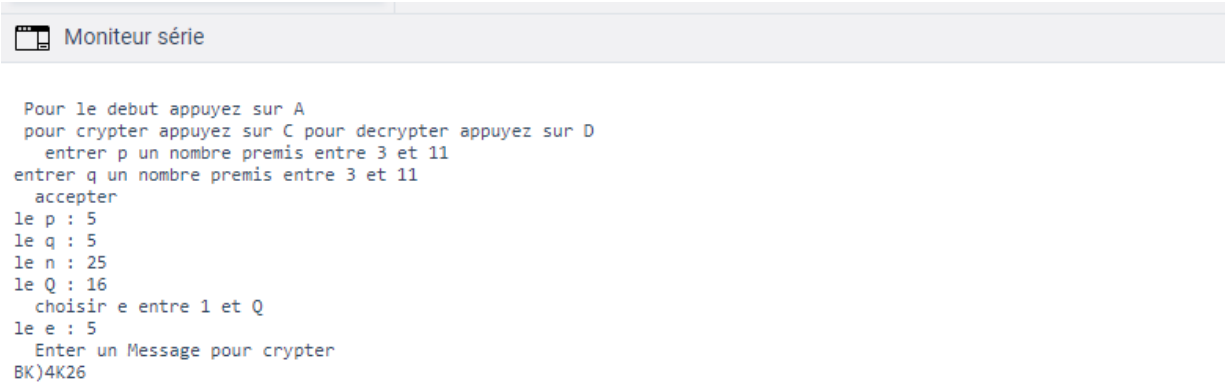
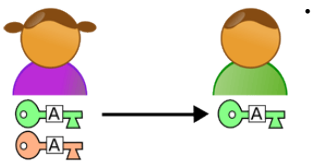
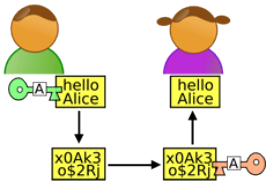


Figure IV.19. Exemple d'exécution
Message : BONJOUR

IV.4.1. Cryptographie à clé publique

Le principe de la cryptographie asymétrique (ou à clé publique) est basé sur l'existence d'une fonction dite à sens unique, pour transformer un message en message codé. Il faut que cette fonction soit simple à appliquer à un quelconque message, mais qu'il soit difficile de retrouver le message original à partir du message codé. La cryptographie à clé publique permet de coder un message secret et aussi d'authentifier l'émetteur d'un message





Le principe de la cryptographie asymétrique (ou à clé publique) est basé sur l'existence d'une fonction dite à sens unique, pour transformer un message en message codé. Il faut que cette fonction soit simple à appliquer à un quelconque message, mais qu'il soit difficile de retrouver le message original à partir du message codé. La cryptographie à clé publique permet de coder un message secret et aussi d'authentifier l'émetteur d'un message. Bob chiffre son message avec la clé publique d'Alice et lui envoie le texte chiffré. Alice déchiffre le message grâce à sa clé privée. **Source : Wikipédia**

IV.4.2. Calcul de temps

On rechiffre les trois exemples précédents avec la même fonction **micros()**

Tableau IV.4. Tableau qui calcule le temps pour RSA

	Temps de Cryptage	**Temps de décryptage**
Paragraphe 1	43,004,448µs	45,234,439 µs
Paragraphe 2	96,162,724 µs	99,161,22 4µs
Paragraphe 3	162,461,248 µs	165,601,200 µs

Remarque :

Une simulation sur Adrduino nécessite une implémentation simple qui se conjugue par des algorithmes moins complexes et qui contient moins de lignes par rapport à un autre développeur comme le Devcpp par exemple.

La figure suivante montre un message d'erreur signalé par une carte Arduino quand elle devient saturée.

Figure IV.20. Message d'erreur

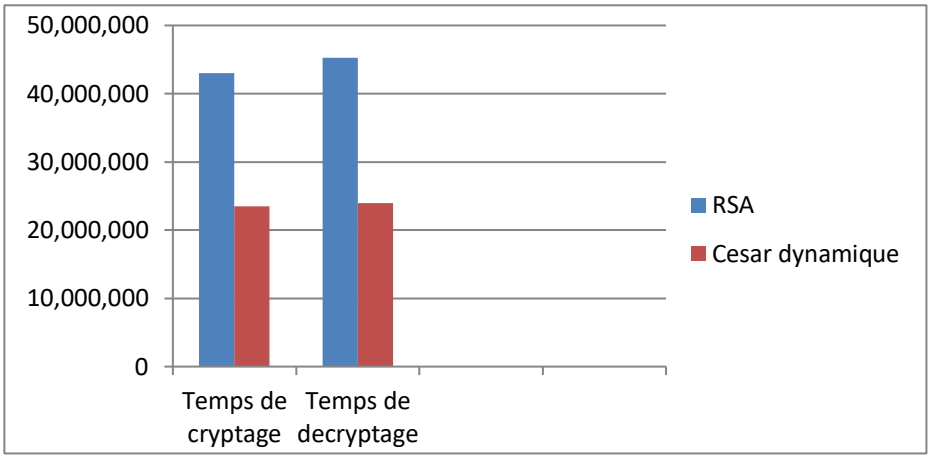


Figure IV.21. Temps de cryptage & décryptage RSA vs César dynamique du paragraphe 1

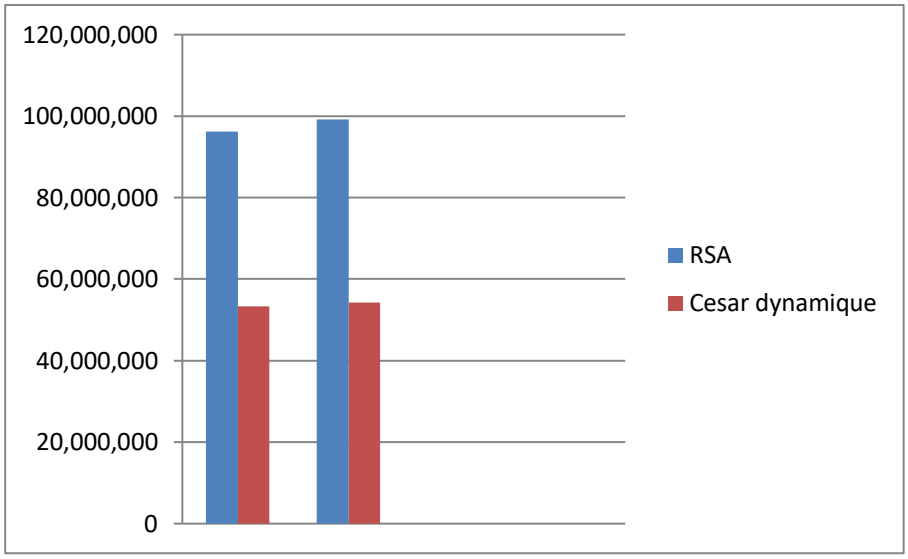


Figure IV.2. Temps de cryptage & décryptage RSA vs César dynamique du paragraphe 2

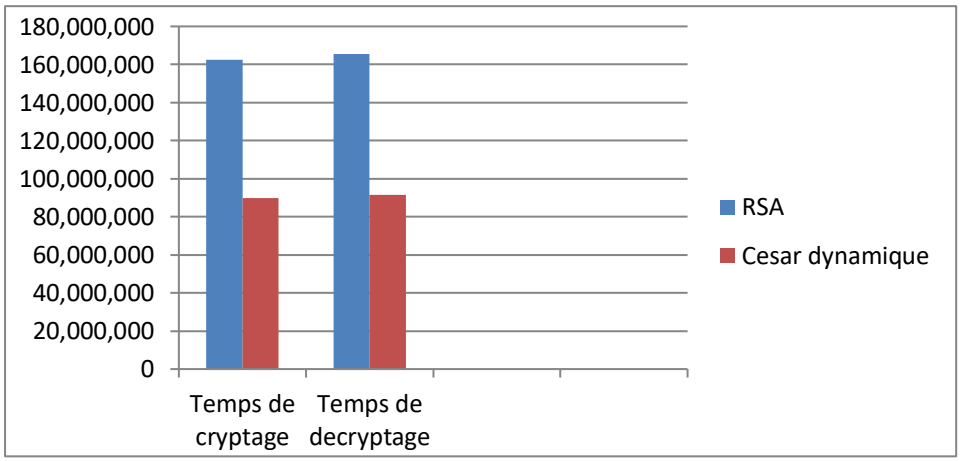


Figure IV.21. Temps de cryptage & décryptage RSA vs César dynamique du paragraphe 3

CONCLUSION :

Dans ce chapitre on élaboré notre travail qui s'est concentré essentiellement sur la simulation, nous avons proposé un concept inédit qui se base sur le chiffrement par César mais en améliorant sa façon de crypter et décrypter les messages. On a évoqué les différents résultats sur les algorithmes qu'on a implémentés sur Arduino. Ça nous a permis de faire une comparaison entre les deux aspects de chiffrement César classique et César dynamique. Nous avons présenté les schémas de simulation sur le site Web gratuit Tinkercad qui est à la base un simulateur et qui permet d'ajouter le circuit Arduino UNO et d'y implémenter notre code source. Ce chapitre contient également des captures d'écrans des différentes exécutions, les codes sources, les tableaux qui calcule le temps de simulation et le nouveau algorithme de cryptographie qu'on a voulu ajouter aussi dans notre travail Système RSA

CONCLUSION GENERALE

Avec les renouveaux des technologies et des réseaux, le taux de présence de l'IOT prend une nouvelle dimension, et devient de plus en plus présent dans plusieurs domaines de vie.

Bien que les besoins soient différents d'un secteur à l'autre, Tout le monde se bénéficie des avantages apportés par l'Internet des objets qui restent assez similaires, on peut citer la vision en temps réel sur tous les processus de production, l'amélioration de la productivité, une prise de décision améliorée …

Comme chaque système dans le monde, la sécurité reste un aspect primordial qui inquiète les utilisateurs et donc les vulnérabilités qui se présentent toucheront bien évidemment plusieurs cotés comme les données de ces objets mais aussi à leurs intégrités et leur authenticités.

L'objectif de cette recherche est de proposer des solutions pour garantir la sécurité de ces objets connectés qui même-si ce sont des systèmes avec des ressources limités et capacités inférieures mais ils resteront souvent menacés et vulnérables aux attaques extérieurs.

Ce présent travail porte sur une découverte d'un sujet d'actualité qui est l'IOT. Nous avons parlé sur la typologie de ces objets ainsi que leur architecture et les technologies dont ils se basent essentiellement. Nous avons cité également les domaines d'application de l'internet of things, ses avantages mais aussi les enjeux et bien essentiellement l'enjeu de sécurité qui reste notre priorité dans cette recherche.

Les objectifs de la sécurité de façon générale sont les mêmes réclamés par le domaine IOT. L'authentification d'un objet est vraisemblablement la preuve de l'identité. La confidentialité, l'intégrité des données, et la répudiation dépendent tous de l'authentification. Un système IOT sans cette approche ne peut pas nous offrir les objectifs de la sécurité convoitée

Notre zone d'étude s'étale sur le concept de la cryptographie désignée comme le système amplement efficace pour garantir la sécurité de nos objets. Pratiquement on a besoin de réaliser une cryptographie légère et comme son nom l'indique on est censé d'effectuer un ensemble des procédés visant à crypter des informations mais de façon légère qui correspond aux caractéristiques de ces objets pour en assurer la confidentialité entre l'émetteur et le destinataire. On s'est focalisé sur l'un des Chiffrement classiques, basiques mais aussi simple et qui arrange nos conditions d'IOT ; C'est le chiffrement par substitution CESAR où les crypto systèmes étaient des algorithmes qui fondent les lettres de l'alphabet.

La dernière partie de ce travail a été consacrée à l'étude des différentes simulations réalisées sur Arduino & Tinkercad après qu'on y a implémenté le code de CESAR et RSA.

La grande découverte dans cette recherche sera le nouveau concept proposé ; c'est celui du chiffrement CESAR mais avec une méthode améliorée et un principe celui du décalage dynamique. Le but de cette idée est bien claire c'est de renforcer notre cryptographie et diminuer la vitesse de casassabilité d'un message chiffré tout en se basant sur la notion de substitution réalisée par les chiffrements anciens de césar.

A la lumière des résultats que nous avons obtenus et des études précédemment réalisées, on détermine des aspects conclues par des comparaisons par rapport a un élément essentiel c'est celui du temps pris lors du traitement divise entre les taches de cryptage et décryptage.

Bibliographie

[1] Evans, D. (2011). The internet of things: How the next evolution of the internet is changing everything. CISCO white paper, 1(2011), 1-11.
[2] Benabdellah, M. (2019). Internet des objets pour l'apprentissage contrôlé en ligne. Revue Internationale d'Economie Numérique, 1(1), 41-49.
[3] L. Atzori, A. Lera, G. Morabioto, The Internet of Things : a survey, Computer Networks 54 (15) (2010) 2787–2805.
[4] D. Miorandi, S. Sicari, F. De-Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks 10 (7) (2012) 1497-1516.
[5] X. Jia, Q. Feng, T. Fan, Q. Lei, RFID technology and its applications in Internet of Things (IoT), 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, 21-23 April 2012, pp. 1282 – 1285.
[6] S. Duquennoy, G. Grimaud, J. J. Vandewalle, The Web of Things: Interconnecting Devices with High Usability and Performance, Zhejiang, 25-27 May 2009, pp. 323 – 330.
[7] G. Aceto , A. Botta , W. Donato , A. Pescapè, Cloud monitoring: A survey, Computer Networks 57 (9) (2013) 2093–2115.
[8] J. Granjal , E. Monteiro, J. Sá Silva, Security in the integration of low power Wireless Sensor Networks with the Internet: A survey, Ad Hoc Networks 24 (2015) 264–287.
[9] Garcia-Morchon, O., et al. "Security Considerations in the IP-based Internet of Things draft-garciacore-security-06." Internet Engineering Task Force (2013).
[10] Garcia-Morchon, O., et al. "Security Considerations in the IP-based Internet of Things draft-garciacore-security-06." Internet Engineering Task Force (2013).

[11] Sahraoui, S. (2016). Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things) (Doctoral dissertation, Université de Batna2). (consulté en Novembre 2015).
[12] Sahraoui, S. (2016). Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things) (Doctoral dissertation, Université de Batna 2). (consulté en Novembre 2015).
[13] Sahraoui, S. (2016). Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things) (Doctoral dissertation, Université de Batna 2). (consulté en Novembre 2015).
[14] Sahraoui, S. (2016). Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things) (Doctoral dissertation, Université de Batna 2). (consulté en Novembre 2015).
[15] Mazaré, M. G. (2007). Gestion à grande échelle de données de capteurs hétérogènes (Doctoral dissertation, Institut National Polytechnique de Grenoble). (consulté en Novembre 2015). [16] Wang, B., Sechilariu, M., & Locment, F. (2012). Intelligent DC microgrid with smart grid communications: Control strategy consideration and design. IEEE transactions on smart grid, 3(4), 2148-2156.
[17] Terir, K. (2020). Gestion de la confidentialité des données pour les dispositifs IOT (Internet of Things) (Doctoral dissertation, University of Jijel).
[18] Llorens, C., Levier, L., Valois, D., & Morin, B. (2011). Tableaux de bord de la sécurité réseau. Editions Eyrolles.
[19] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 17(4), 2347-2376.
[20] Garcia-Morchon, O., et al. "Security Considerations in the IP-based Internet of Things draft-garciacore-security-06." Internet Engineering Task Force (2013)
[21] Schneier, B. (2016). Lessons from the dyn ddos attack. Schneier on Security, 8.

[22] Tankard, C. (2015). The security issues of the Internet of Things. Computer Fraud & Security, 2015(9), 11-14..
[23] Jerkins, J. A. (2017, January). Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code. In 2017 IEEE 7th annual computing and communication workshop and conference (CCWC) (pp. 1-5). IEEE.
[24] Gayraud, V., Nuaymi, L., Dupont, F., Gombault, S., & Tharon, B. (2003, June). La sécurité dans les réseaux sans fil ad hoc. In Actes du symposium SSTIC03.
[25] Efremov, A. A., & Bessonova, C. E. (2015). " Smart Home" Risk Analysis. Вестник полиции, (2), 48-54.
[27] Md Mahmud Hossain, MaziarFotouhi, and Ragib Hasan. Towards an analysis of security issues, challenges, and open problems in the internet of things. In Services (SERVICES), 2015 IEEE World Congress On, pages 21–28. IEEE, 2015.
[28] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, ElieBursztein, Jaime Cochran, ZakirDurumeric, J Alex Halderman, Luca Invernizzi, Michalis Kallitsis, et al. Understanding the mirai botnet. In USENIX Security Symposium, 2017.
[29] Ben Herzberg, Dima Bekerman, and IgalZeifman. Breaking down Mirai : An IoT DDoS Botnet Analysis. 2016.
[30] Krebs, B. (2014). Target hackers broke in via HVAC company. Krebs on Security, 5.
[31] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. Fog Computing : A Platform for Internet of Things and Analytics. In Nik Bessis and CiprianDobre, editors, Big Data and Internet of Things : A Roadmap for Smart Environments, volume 546, pages 169–186. Springer International Publishing, 2014.
[32] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and SateeshAddepalli.Fog computing and its role in the internet of things. In Proceedings of the First

Edition of the MCC Workshop on Mobile Cloud Computing - MCC ’12, page 13. ACM Press, 2012.
[33] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things : Architecture, enabling technologies, security and privacy, and applications. 4(5) :1125–1142, 2017.
[34] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. Security in the internet of things : A review. In 2012 International Conference on Computer Science and Electronics Engineering, volume 3, pages 648–651. IEEE, 2012.
[35] De Vaujany, F. X., Bohas, A., Fabbri, J., & Laniray, P. (2016). Nouvelles pratiques de travail: La fin du clivage salariat-entrepreneuriat? (Doctoral dissertation, Research Group on Collaborative Spaces).
[36] Lara-Nino, Carlos Andres, Arturo Diaz-Perez, and Miguel Morales-Sandoval. "Elliptic curve lightweight cryptography: A survey." IEEE Access 6 (2018): 72514-72550.
[37] Chatzigiannakis, Ioannis, et al. "Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices." 2011 IEEE eighth international conference on mobile ad-hoc and sensor systems. IEEE, 2011.
[38] Azizi, Abdelmalek. Extraits de l'Histoire de la Cryptographie au Maroc. Diss. Cryptography, 2009.
[39] Mrayati, M., Y. Meer Alam, and H. Al-Tayyan. "Arabic Origins of Cryptology." Volumes 1.2 (2003): 3.
[40] Wurm, Jacob, et al. "Security analysis on consumer and industrial IoT devices." 2016 21st Asia and South Pacific design automation conference (ASP-DAC). IEEE, 2016.
[41] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, “Elliptic Curve Lightweight Cryptography: a Survey,” IEEE Access, vol. PP, no. c, pp., 2018.
[42] S. Chandra, S. Paira, S. S. Alam, and G. Sanyal, “A comparative survey of symmetric and asymmetric key cryptography,” 2014 Int. Conf. Electron. Commun. Comput. Eng. ICECCE 2014, pp. 83–93, 2014.

[43] Pinol, Oriol Pinol, et al. "BSD-based elliptic curve cryptography for the open Internet of Things." 2015 7th International Conference on New Technologies, Mobility and Security (NTMS). IEEE, 2015.
[44] B. Schneier, Cryptographie appliquée. 2ième édition, 1996.
[45] Schwer, Sylviane, and Jean-Michel Autebert. "Henri-Auguste Delannoy, une biographie (1e partie)." Mathématiques et sciences humaines. Mathematics and social sciences 174 (2006): p-25.
[46] Amblard, Zoé. Cryptographie quantique et applications spatiales. Diss. Université de Limoges, 2016.
[47] Labouret, Ghislaine. "Introduction à la cryptographie." Support du cours du cabinet Hervé Schauer (HSC). France (2001)..
[48] Malalatiana, Ramafiarisona Hajasoa, and Randriamitantsoa Paul Auguste. "Security Transfect of Image Numeric by Cryptotattoing With Vernam-Multiresolution Analysis Combinate." (2019).
[49] Levenda, Peter. "Post new topic Reply to topic 9/11, 7/7, Covid-1984 & the War on Freedom Forum Index-> The Bigger Picture.".
[50] Martin, Bruno. Codage, cryptologie et applications. PPUR presses polytechniques, 2004.
[51] Yasser, Bousnoubra, and Hamada Aymen. "La cryptographie des images numériques par la carte logistique chaotique." (2020).
[52] Guillou, Louis C., Marc Davio, and Jean-Jacques Quisquater. "L'état de l'art en matière de techniques à clé publique." Annales des télécommunications. Vol. 43. No. 9. Springer-Verlag, 1988.