MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université Saad Dahlab de Blida 1

Faculté des sciences

Département d'informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master en Informatique

OPTION : SECURITE DES SYSTEMES D'INFORMATION

THEME :

Etude et Implémentation d'un Mécanisme de Sécurité Contre les intrusions pour le Routage Centré Contenu

Réalisé par

LATRECHE FATMA ZOHRA

Devant le jury composé de :

-	**Mr Benyahia Mohamed**	**MAA**	**Président**
-	**Mme Arkam Meriem**	**MAA**	**Promotrice**
-	**Mme Djeddar Afrah**	**MCB**	**Examinatrice**

Année Universitaire : 2018-2019

RESUME

Le réseau centré sur l'information (Information Centric Networking) est un nouveau paradigme de communication qui met l'accent sur la récupération de contenu à partir du réseau, indépendamment de l'emplacement, du stockage ou de la représentation physique de ce contenu. En ICN, la sécurité du contenu est très importante vu les attaques déjà recensées dans la littérature à savoir : DDOS, Blocage mobil, interception…etc. Dans notre travail nous nous intéressons aux attaques d'interception (Spoofing) pour cela nous nous sommes basées d'abord sur l'étude des réseaux (ICN), par la suite les mécanismes utilisés pour les attaques d'interceptions avant d'étudier les contres mesures existantes dans la littérature. En faisant cette étude nous avons simulé un scénario d'attaque en utilisant le package CCN-lite sous le simulateur OMNeT++, par la suite nous avons introduit notre encapsulation chiffrée pour sécurisé l'ICN.

Mot clé: réseaux centrés sur l'information (ICN), routage centré contenu, sécurité, spoofing, interception, CCN-lite, encapsulation chiffrée.

ABSTRACT

The Information Centric Networking is a new communication paradigm that focuses on recovering content from the network, regardless of the location, storage, or physical representation of that content. In ICN, content security is very important given the attacks already identified in the literature, namely: DDOS, mobile blocking, interception ... etc. In our work we are interested in interception attacks (Spoofing) for that we based ourselves first on the study of the networks (ICN), later the mechanisms used for the attacks of interceptions before studying the against existing measures in the literature. In doing this study we simulated an attack scenario using the CCN-lite package under the OMNeT ++ simulator, after which we introduced our encrypted encapsulation to secure the ICN.

Keyword: information-centric networks (ICN), content-centric routing, security, spoofing, interception, CCN-lite, encrypted encapsulation.

ملخص

شبكة المعلومات المركزية هي نموذج اتصال جديد يركز على استعادة المحتوى من الشبكة ، بغض النظر عن الموقع أو التخزين أو التمثيل الفعلي لهذا المحتوى. في ICN ، يعد أمان المحتوى مهمًا جدًا نظرًا للهجمات التي تم تحديدها بالفعل في الأدبيات ، وهي : DDOS ، وحظر الهاتف المحمول ، والاعتراض ... إلخ. في عملنا ، نحن مهتمون بهجمات الاعتراض (الخداع) لأننا اعتمدنا أولاً على دراسة الشبكات (ICN) ، فيما بعد الآليات المستخدمة لهجمات الاعتراض قبل دراسة ضد التدابير القائمة في الأدب. في هذه الدراسة ، قمنا بمحاكاة سيناريو الهجوم باستخدام حزمة CCN-LITE تحت محاكاة OMNET++ ، وبعد ذلك قدمنا التغليف المشفر لتأمين ICN.

الكلمة الأساسية: الشبكات المتمركزة على المعلومات (ICN) ، والتوجيه المتمحور حول المحتوى ، والأمن ، والخداع ، والاعتراض ، CCN-LITE ، التغليف المشفر.

Remerciement

Je remercie d’abord ALLAH le tout puissant qui m’a guidé et qui m’a donné la force et la volonté de réaliser ce travail.

Mes pensées vont vers mes parents, qui ont toujours cru en moi.

C’est grâce à leur soutien que j’ai pu réaliser ce travail. Ils savent déjà combien je leur dois.

Comme je remercie ma promotrice Mme Meriem ARKAM de m’avoir pris en charge et aidé tout au long du projet.

Mes remerciements les plus sincères à toutes les personnes qui avaient contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Enfin, je tiens aussi à remercier les membres du jury d’avoir accepté d’examiner et de juger mon travail.

Merci à tous et à toutes.

Table des Matières

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE D'ACRONYMES

Acronyme	Intitulé
ICN :	Information Centric Networking
DDOS :	Denial of Service attack
OMNeT++ :	**O**bjective **M**odular **Ne**twork **T**estbed in C++
CCN-lite :	Convention Collective Nationale lite
DONA :	Data Oriented Network Architecture
NGI :	New Generation Internet
NetInf :	Network of Information
PSIRP :	Publish Subscribe Internet Routing Paradigm
NDN :	Naamed Data Network
xDSL :	Digital subscriber line
TCP/IP :	Transmission Control Protocol/Internet Protocol
WSN :	Wirless Sensor Network
WIFI :	Wirless Fidelity
3G/4G :	Troisième Génération / quatrième Génération
URL :	Uniform Resource Locator
SAIL :	Software Architecture Integration Library
4WARD :	Forward
COMET :	Concurrent Object Modeling and Architectural Design Method
CONET :	Cooperative Wireless Communications and Networking
UC :	University College
NN(RN) :	Nœud de rendez-vous
DHT :	Destributed hash table

FSN :	Science Foundation
CS :	Content Store
PIT :	Pending Interest Table
FIB :	Forwarding Information Base
PKI :	Public Key Infrastructure
RSA :	Ronald Rivest, Adi Shamir et Leonard Adleman
INET :	Institut National des Etudes Territoriales
PARC :	Palo Alto Research Center
IoT :	Internet Of Things

INTRODUCTION GENERALE

Selon le Cisco Visual Networking Index 2016 et d'ici 2020 [1], il y aura près de 4,1 milliards d'utilisateurs Internet et 26,3 milliards d'appareils réseau et de connexions dans le monde, le débit moyen des connexions à large bande fixe augmentera à 47,7 Mbps et la vidéo IP à 82%. De tout le trafic. Cette demande croissante de distribution de contenus hautement évolutive et efficace nécessite de nouvelles solutions alternatives pour le futur Internet de prochaine génération (Next Generation Internet NGI), l'architecture Internet existante devenant de plus en plus inadéquate. Le réseau centré sur l'information (ICN) est l'une des alternatives de NGI, qui se concentre sur le contenu plutôt que sur les points d'extrémité. ICN s'appuie sur des attributs uniques tels que la dénomination indépendante de l'emplacement, la mise en cache sur le réseau, le routage basé sur le nom et la sécurité intégrée. En parlant du routage, les projets ICNs mènent à plusieurs défis pour la protection du contenu des différentes attaques citées dans la littérature tout en se mettant en concurrence selon le niveau de sécurité, les coûts et la latence. Pour atteindre les objectifs de sécurité souhaitables, il est nécessaire d'identifier les problèmes et les failles des différents attributs du réseau à exploiter dans les attaques, ce qui va aider par la suite à cerner les objectifs à atteindre dans les solutions à proposer et afin de permettre un très bon degré de sécurité. Notre travail sera organisé en trois chapitres en commençant par la présentation des réseaux ICNs, leur principe de fonctionnement ainsi que de faire une comparaison entre ces projets les plus connus, en passant au deuxième chapitre nous allons parler de la sécurité, de quelques attaques connues par lesquelles l'ICN est menacé ainsi de leur classification et leurs impactes sur la sécurité du réseau, ça n'empêche pas de définir les avantages de sécurité de cette architecture, par la suite nous allons détailler une des attaques du routage qui est l'interception, les failles à exploiter pour en survenir, son déroulement ainsi que de quelques solutions qui ont été proposés précédemment. Arrivé au dernier chapitre, nous allons définir notre approche pour atténuer le risque de l'attaque précédemment étudié, et pour donner sens à notre travail nous allons implémenter notre solution tout en citant les étapes suivies. Enfin nous terminons avec une conclusion générale et perspectives.

CHAPITRE I : RESEAUX CENTRES SUR L'INFORMATION

1. Introduction

La formidable croissance d'Internet au cours de la dernière décennie a apporté des applications innovantes et interactives (multimédia, jeux en ligne et informatique en nuage), ainsi que des exigences strictes pour les utilisateurs: environnement de distribution de contenu, mobilité, omniprésence, sécurité et confiance…etc. a conduit à un changement de paradigme d'une approche de réseau centrée sur l'hôte à une approche axée sur l'information, car aujourd'hui, les utilisateurs sont davantage intéressés par le contenu que par l'emplacement des nœuds homologues en communication.

En commençant par ce chapitre, nous allons aborder les réseaux centrés sur l'information. Par la suite, nous allons voir les principaux projets ICNs (DONA, NetInf, PSIRP et CCN/NDN), ainsi qu'une comparaison entre eux, par la suite nous ferons une description détaillée sur le routage dans les CCNs. Nous finissons par présenter les problèmes de sécurité de cette nouvelle architecture, ainsi que les solutions de sécurité offertes par l'architecture CCN.

2. Les réseaux centrés sur l'information

Internet a été développé sur la base de l'architecture orientée hôte (voir Figure 1) et prend en charge différents types de communication. En raison de la large utilisation de l'Internet actuel, le besoin d'adresses IP s'est beaucoup accru pour identifier plusieurs hôtes. IPv6 a été développé pour résoudre la pénurie d'adresses IPv4 bien qu'il ne soit pas largement utilisé comme prévu [2].

Mais au fur et à mesure, Internet est devenu de plus en plus populaire et de plus en plus utilisé, ce qui a conduit à une évolution d'Internet et des technologies, les réseaux étant plus complexe qu'avant. Les terminaux ne sont plus simplement des ordinateurs bureautiques ou portables, mais sont maintenant des smart phones, des tablettes, des terminaux de jeux vidéo… etc. Les infrastructures réseaux ne se limitent plus à xDSL ou à la fibre optique; les réseaux WiFi, mobile 3G/4G, WSN et satellites sont maintenant largement déployés. La structure de TCP/IP et la couche de transport devient de plus en plus complexe pour gérer cette pluralité d'environnements [3].

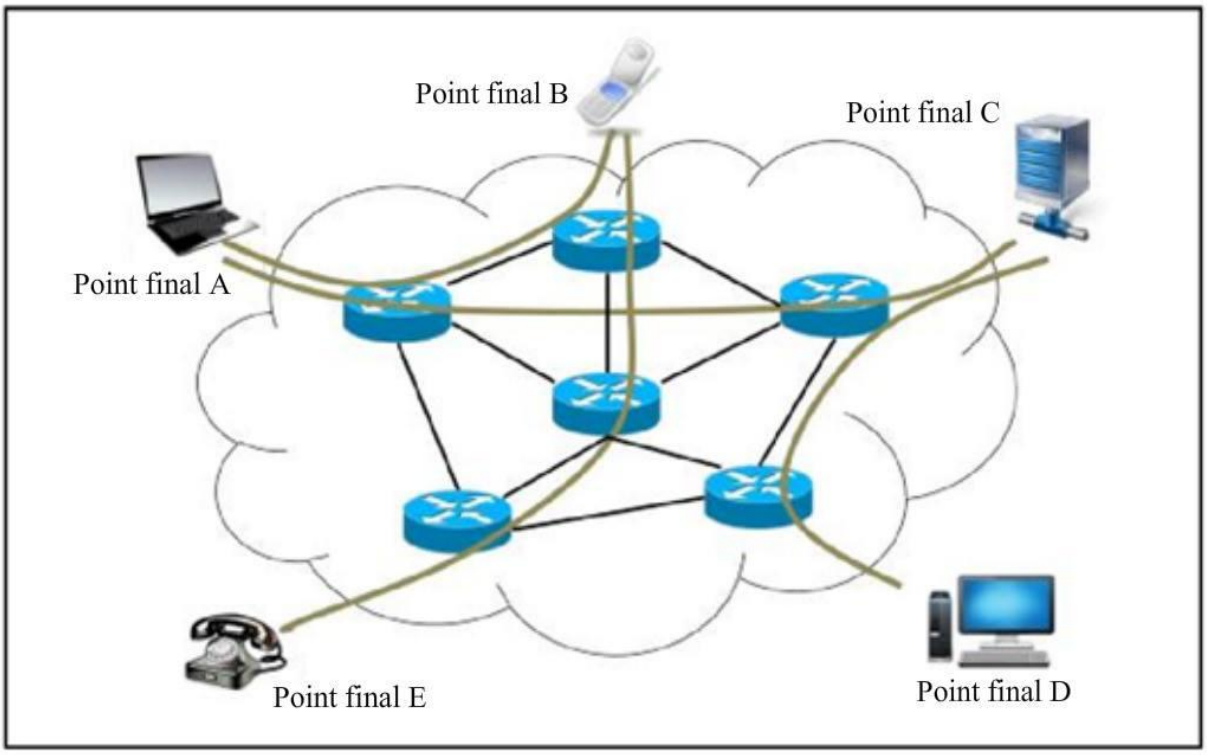

Figure 1: L'architecture de réseaux IP [3]

Dans cet aspect, le réseau centré sur l'information (ICN) apparaît comme une approche potentielle qui introduit les données nommées indépendantes de l'emplacement et de l'application en tant que principe fondamental de la diffusion et de la récupération du contenu tout en suivant l'approche axée sur le récepteur (voir Figure 2). Un utilisateur accède aux données en envoyant un paquet de requête contenant le nom du contenu sans avoir à connaître l'emplacement du contenu.

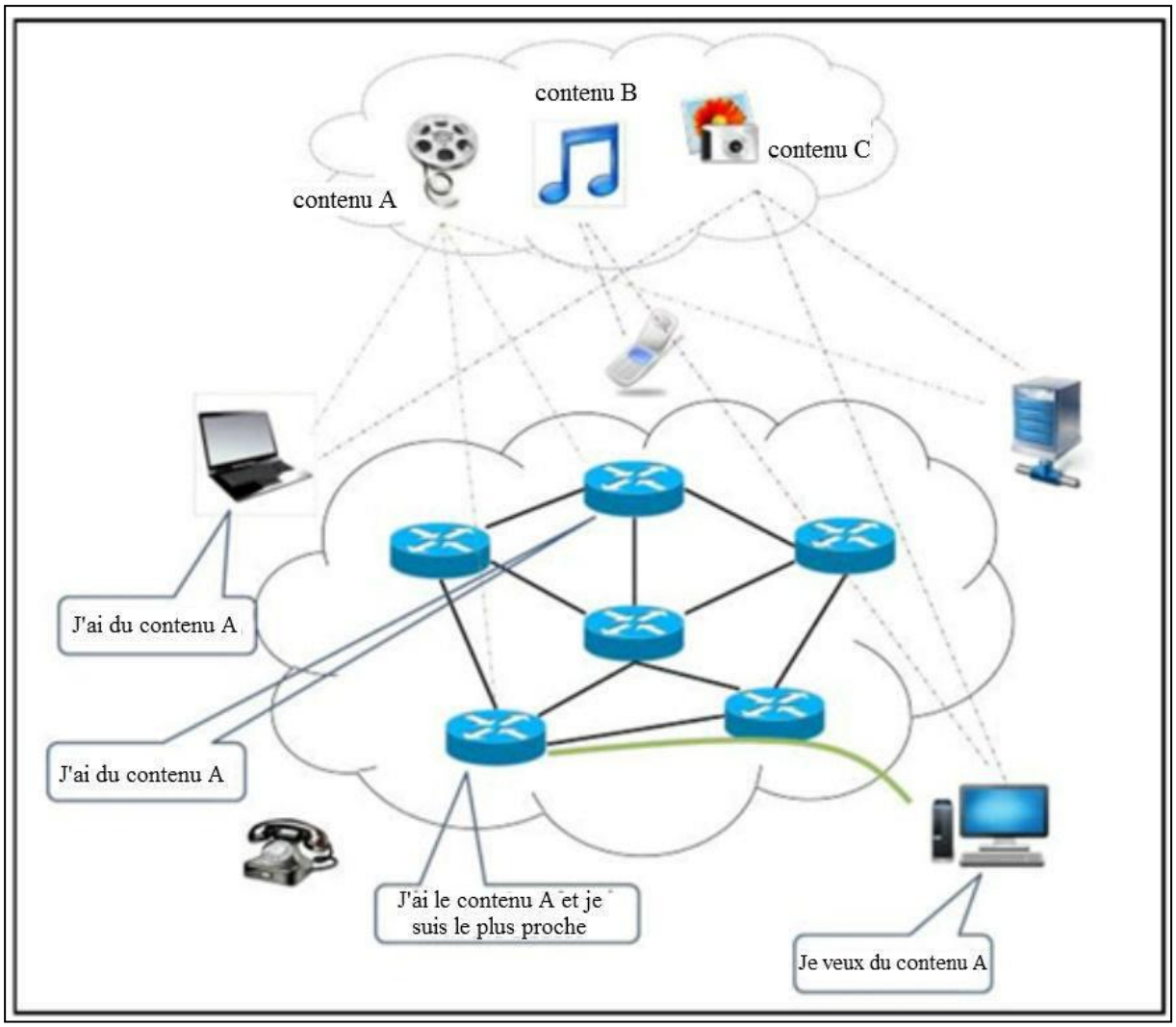

Figure 2: L'architecture des réseaux ICNs [3]

ICN promet d'améliorer l'efficacité en termes d'évolutivité, de consommation de bande passante et de robustesse. L'un des objectifs principaux d'ICN est de refléter les besoins actuels et futurs des architectures existantes, axés sur l'utilisateur. [4]

2.1. Fonctions de base des ICNs

Les ICNs proposent de changer l'Internet actuel, qui est basé sur les localisations des serveurs avec des adresses bien définies, vers une architecture basée sur le nom des contenus, avec des fonctionnalités nativement intégrées comme le nommage indépendant de la localisation, le routage basé sur les noms de contenu, la faculté de cacher des contenus dans les réseaux, le multicast, la sécurisation des contenus, la mobilité…etc. Grâce à ces fonctionnalités, les ICNs sont plus efficaces pour délivrer des contenus aux utilisateurs avec

une meilleure qualité et permettent aussi d'améliorer la gestion des capacités réseaux des fournisseurs des réseaux. [3]

Toutes les architectures ICN ont des concepts génériques, qui peuvent être classés comme suit: objet d'information, dénomination, routage, mise en cache, sécurité et interface de programmation d'application. L'objet information se rapporte au contenu lui-même, qui est l'objectif principal d'ICN, quel que soit son emplacement de stockage et de représentation physique. Pour chaque contenu, il peut y avoir différentes représentations et différentes copies pour chaque représentation.

Voici les principales fonctionnalités intégrées dans un ICN :

2.1.1. Nommage des contenus

Les schémas de nommage dans ICN peuvent être classés en trois catégories: plate, hiérarchique ou hybride. Dans un ICN, l'unité de réseau de base est l'objet de contenu sur lequel toutes les activités de réseautage sont basées, Un objet de contenu est identifié par un nom globalement unique qui est composé d'un nombre variable de composants, organisés dans l'une des structures plates, hiérarchiques ou hybride que nous allons voir après.

2.1.2. Routage basé sur le nom de contenu

Le réseau ICN est un modèle basé sur le récepteur. C'est à-dire qu'un utilisateur exprime seulement son intérêt sur des contenus au réseau. Ensuite, c'est le réseau qui a pour charge de trouver les bons contenus et les meilleures sources pour ces contenus, en se basant sur leurs noms. Quand l'intérêt du client arrive finalement à une source de contenu, le contenu est délivré en suivant le chemin inverse du message d'intérêt jusqu'au client. Au final, le client est satisfait car il reçoit le contenu désiré, même s'il n'a pas connaissance de l'entité qui lui a fourni ce contenu. [3]

En suivant notre objectif nous allons passer par la suite à la sécurité liée au routage.

2.1.3. Cache des contenus

La mise en cache dans le réseau dans ICN respecte les principes suivants: [5]

- ***Uniforme**:* c'est-à-dire appliqué à tout le contenu transmis par n'importe quel protocole.
- ***Démocratique:*** c'est-à-dire publié par n'importe quel fournisseur de contenu.
- ***Omniprésente:*** c'est-à-dire disponible pour tous les nœuds de réseau.

C'est une caractéristique importante des ICNs De sorte que les demandes ultérieures pourront être satisfaites plus rapidement, directement par les caches des nœuds ICN. Les paquets perdus pourront aussi être récupérés plus rapidement par des retransmissions directes depuis les caches les plus proches [3]**.**

2.1.4. Sécurité du contenu

De nouveaux concepts de sécurité centrés sur l'information ont été mis tel que sécurité auto-protégée via des contenus cryptés et auto certifiés, et non pas via des connexions de communication sécurisées comme IP. Seuls les utilisateurs autorisés peuvent déchiffrer les contenus. Il est nécessaire pour que la sécurité soit appliquée au contenu lui-même. [3]

2.1.5. Interface de programmation d'application (API)

Une API dans ICN est utilisée pour demander et fournir le contenu. La source publie son contenu pour le rendre disponible pour les autres utilisateurs du réseau. [5]

2.1.6. Mobilité du contenu

La mobilité des utilisateurs n'influe pas le comportement des réseaux ICN. Leurs demandes, issues de différents endroits à différents moments, sont traitées indépendamment par les réseaux ICN, chacune comme une requête unique. D'où les chercheurs tentent de réaliser la 5G.

2.2. Architectures d'ICN

Presque toutes les architectures ICN proposent un modèle de récupération de données piloté par le récepteur. Nous divisons les architectures ICN existantes en trois catégories, à savoir les architectures sans structure ou à base plate, structurées ou hiérarchiques et hybrides. Dans ce qui suit, nous discutons de ces catégorisations en détail. [4]

2.2.1. Architectures ICN à base plate

Dans cette catégorie, des chaînes plates, indépendantes du lieu et uniques au monde sont utilisées pour l'identification du contenu, et les architectures n'ont pas de structure pour

l'attribution de noms de contenu. Plusieurs types de noms à plat sont utilisés dans les implémentations existantes telles que DONA, PSIRP, CDN et MobilityFirst…etc. Ces architectures attribuent des noms à plat au contenu, puis ces noms sont diffusés dans le réseau. Le destinataire demande le contenu en utilisant le nom du contenu qui est résolu au fur et à mesure que la demande avance vers la source. Bien que ces architectures simplifient la gestion du réseau (c'est-à-dire le besoin d'administrateurs), elles posent des problèmes d'évolutivité en termes de routage car le contenu doit être mis en cache au niveau des routeurs. Un des principaux défis est d'assurer l'unicité mondiale des noms afin que les demandes puissent être traitées efficacement. [4]

2.2.2. Architectures ICN basées sur la hiérarchie

Dans les architectures ICN basées sur la hiérarchie, le contenu est attribué à des noms hiérarchiques indépendants de l'application et du lieu, qui sont généralement similaires aux URL sur Internet. Cette approche est considérée comme plus évolutive car une structure hiérarchique complète est définie, ce qui facilite le routage d'une demande de contenu et est plus attrayante en raison des mécanismes de transmission basés sur les préfixes, ce qui la rend plus efficace. Cependant, le principal défi consiste à définir et à maintenir des structures hiérarchiques utiles**.** [4]

2.2.3. Architectures ICN hybrides

Dans cette catégorie, il est possible d'affecter des noms au contenu, les noms comportent un mélange d'architecture plate et hiérarchisés. [4]

2.3. Principaux projets d'ICN

Dans cette sous-section, nous examinons certaines architectures ICN représentatives, notamment DONA, CCN/NDN, PSIRP / PURSUIT et NetInf. Nous renvoyons les lecteurs intéressés à deux enquêtes pour plus de détails sur d'autres architectures ICN, telles que SAIL, 4WARD, COMET, CONVER-GENCE et CONET. Dans cette étude, nous nous intéresserons plus particulièrement à trois architectures: CCN/NDN, PSIRP / PURSUIT et NetInf. Ces trois ont reçu l'attention de la communauté dans le passé et continuent d'être

favorisées en tant qu'architecture de choix. [6]

2.3.1. L'architecture de réseau orientée données (DONA)

Data Oriented Network Architecture (DONA) une architecture qui a été proposée par Koponen et al. chez UC Berkeley en 2007 [6]. DONA utilise un système de nommage plat auto-certifiant. Chaque nom est composé de deux parties. Le premier est le hachage cryptographique de la clé publique de l'éditeur et le second est un identificateur d'objet, attribué par l'éditeur et unique dans le domaine de l'éditeur.

Le service de résolution de DONA est composé d'un réseau d'entités de gestionnaire de résolution (RH), interconnectées hiérarchiquement, chargées de la publication et de la récupération des objets.

Pour publier un objet (voir figure 3), le propriétaire envoie un message REGISTER incluant le nom de l'objet à son RH locale. L'hôte locale transmet ce message à ses partenaires parents et homologues, qui stockent ensuite un mappage entre l'adresse de l'agent local et le nom de l'objet. Un abonné intéressé par l'objet envoie un message FIND avec le nom de l'objet à son propre agent RH local. La RH locale transmet cette demande à sa mère RH. La propagation continue jusqu'à ce qu'une correspondance soit trouvée quelque part dans la hiérarchie. Après avoir trouvé une correspondance, la demande est transmise à l'éditeur identifié. Les auteurs livrent l'objet d'un éditeur à un demandeur à l'aide du réseau IP sous-jacent ou par le parcourt du chemin inverse de l'éditeur au demandeur. [6]

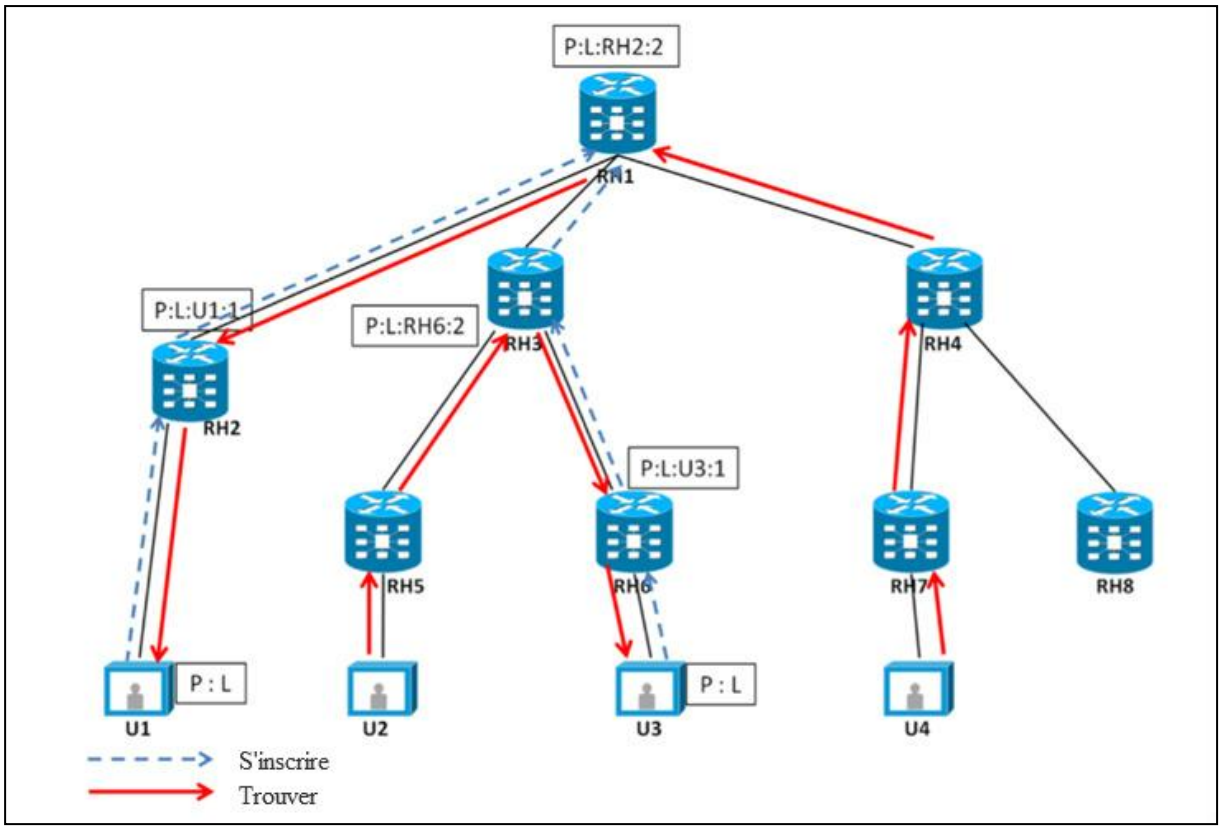

Figure 3: Le schéma de routage de DONA. [3]

2.3.2. Le projet Publish Subscribe Internet Routing Paradigm (PSIRP)

Publish Subscribe Internet Routing Paradigm (PSIRP) est un réseau composé de trois entités principales (voir figure 4), à savoir les nœuds Rendez-vous NN (RN) qui forment le réseau Rendez-vous (RENE), le gestionnaire de topologie et les expéditeurs. Semblable à DONA, PURSUIT utilise un nom simple, composé d'un identifiant de portée, qui regroupe des informations connexes, et un identifiant de rendez-vous, qui garantit que l'identifiant de chaque objet est unique dans son groupe. [6]

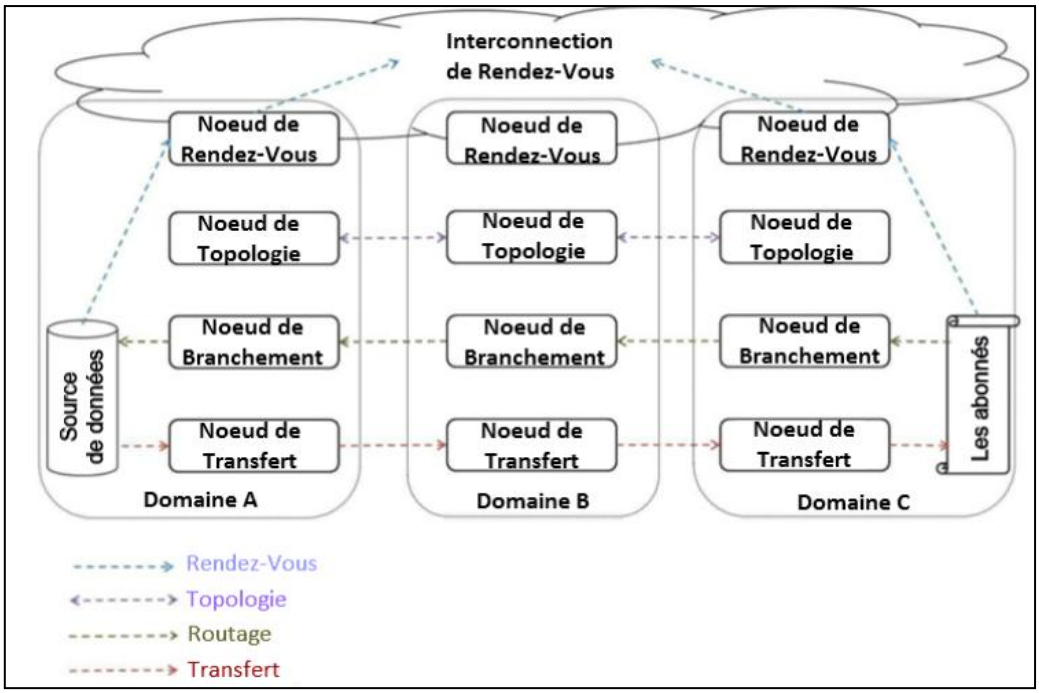

Figure 4: Schéma de routage PSIRP [3]

Un éditeur fait de la publicité pour son contenu en envoyant un message PUBLISH à son RN local (le RN à proximité de l'éditeur), qui le dirige vers le RN désigné pour stocker le nom du contenu défini par le télescope (désigné par RN). La RN locale prend cette décision en utilisant une table de hachage distribuée (DHT). Un abonné intéressé par l'objet de contenu envoie un message SUBSCRIBE à son RN local, qui sera également acheminé vers le RN désigné à l'aide du DHT.

2.3.3. Le réseau d'information (NetInf)

Network Information a été initialement conçu dans le cadre du projet FP7 4WARD. NetInf utilise un schéma de dénomination à plat avec une liaison entre les noms et leurs localisateurs, qui pointent vers l'emplacement du contenu. Étant donné que plusieurs nœuds peuvent mettre en cache des copies des données, un objet peut être lié à plus d'un unique localisateur. [6]

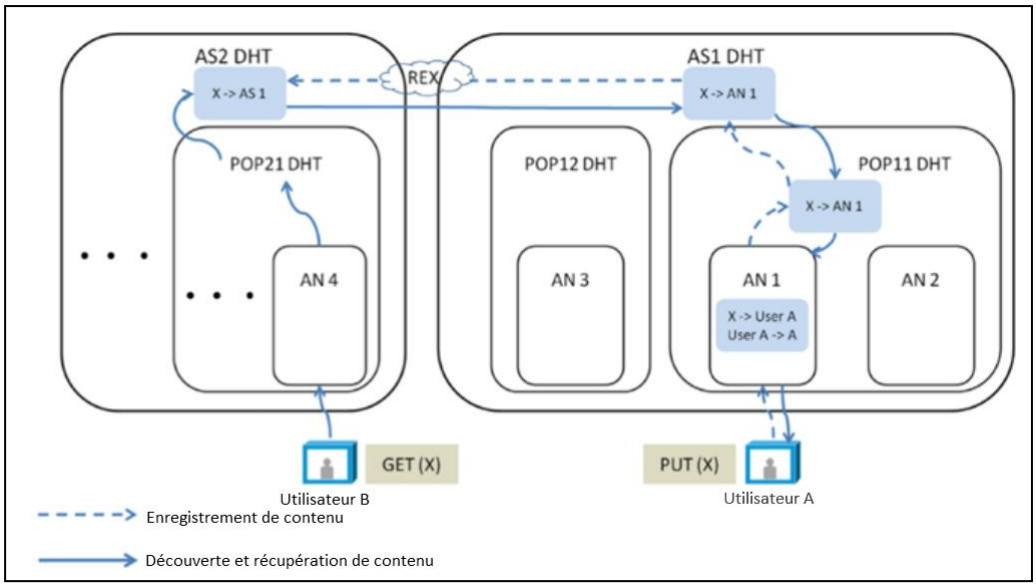

Figure 5: Le schéma de routage de NetInf [3]

NetInf propose un modèle de routage basé sur le nom, un client envoie directement un message GET avec le nom du data object (voir figure 5). Ce message est transféré vers un stockage disponible utilisant un routage basé sur le nom et l'objet de données, une fois trouvé il sera transféré au client.

2.3.4. Les réseaux centrés sur le contenu (CCN /NDN)

Les Réseaux Centrés sur le Contenu ont été proposés par des chercheurs du centre de recherche de Palo Alto en 2009. En 2010, le réseau américain Named (Named) [6], qui suit les principes de la conception nommée, à Science Foundation (NSF) est l'un des quatre projets à financer dans le cadre du programme Future Internet Architecture de la FSN. Les noms CCN et NDN partagent les mêmes principes fondamentaux, tels qu'un schéma de nommage hiérarchique, la mise en cache du contenu et le routage de contenu nommé (le nom de domaine (NDN) était CCN avant sa création).

La dénomination hiérarchique permet au nom de domaine du fournisseur d'être utilisé dans la prise de décisions de routage.

La figure 6 montre un exemple du schéma de dénomination dans lequel un fichier vidéo est identifié par le nom **/parc.com/ video/ WidgetA.mpg/ _v<timestamp>/** et ses morceaux avec des noms du type **/parc.com/ video/ WidgetA.mpg/ _v<timestamp>/ _s_id** où **_s_id** représente l'identifiant du segment (paquet de données).

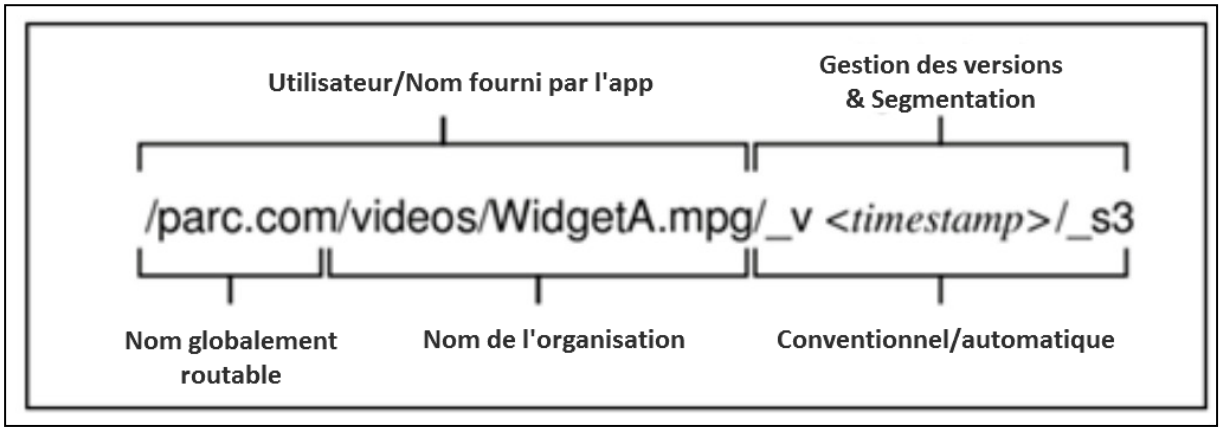

Figure 6: Exemple de nom hiérarchique de CCN/NDN. [16]

De plus, l'architecture du CCN/NDN permettrait une recherche simple et directe d'un contenu sans avoir à identifier son détenteur, comme dans le cas des réseaux IP. Un utilisateur cherche une donnée à travers son nom. Dès lors que la requête est lancée, une demande sous forme d'un paquet dit "Intérêt" est envoyée à son routeur d'accès. Si la donnée n'est pas présente dans le Content Store (CS) de ce routeur, la requête se propage au fur et à mesure dans le réseau. Une fois la donnée trouvée, elle suit le chemin inverse de la requête de recherche jusqu'à l'utilisateur final et sera stockée dans un CS dans les routeurs CCN intermédiaires. Cette architecture offre plusieurs possibilités de disponibilité indépendamment de l'adresse d'une machine. La sécurité est associée directement aux données et pas aux "conteneurs" (liens, routeurs, serveurs…etc.) ce qui permet d'ajuster de manière très flexible le niveau de sécurité à la nature du contenu en question. Plus intéressant encore, les contenus ne sont plus associés à des conteneurs précis mais peuvent être dupliqués à volonté et stockés notamment dans des mémoires caches au sein du réseau. On discutera en détail le fonctionnement et le routage de CCN. [6]

2.3.5. Comparaison

Dans le tableau suivant, nous comparons les 4 projets selon les fonctions de base ci-dessus. [18]

	DONA	**NetInf**	**PSIRP**	**CCN/NDN**
Début/fin	-2007	-2010/2013	-2008/2010	-2009

de projet				
Nommage	-Plat	-Plat	-Plat	-Hiérarchique
Routage	-Basé sur le nom	-Basé sur le nom -Résolution des noms	-Basé sur le nom -Utilisation du modèle de résolution « point de rendez-vous	-Basé sur le nom
Cache	-Les RHs peuvent être activés avec un mécanisme de mise en cache universel	-mise en cache sur le chemin -mise en cache hors chemin -mise en cache des pairs	-Les caches multiples d'un objet peuvent être maintenues en fonction de la portée du point de rendez-vous pour l'identifiant associé à l'objet	-La mise en cache du contenu dans le CS d'un nœud -Le stockage des paquets est possible à chaque nœud NDN
Mobilité	-Protocole de transport TCP -Fournit des mécanismes d'acheminement et de transport	-Mécanismes de transmission en mode message de demande/réponse	-Transfert d'objet à partir du nom d'origine(unique) -Une bonne géstion de contrôle de flux	-Aucune fonctionnalité de couche de transport -Fournie par l'application ou des bibliothèques de support
Sécurité	-L'espace de noms autocertifié a pour l'intégrité des	- L'espace de noms autocertifié -sécurité des objets fournie par la cryptographie de clé publique	-L'espace de noms auto-certifié. -cryptographie à courbe elliptique (ECC) pour vérifier	-Signature cryptographique des paquets de données par l'éditeur

	noms-données		la signature et l'authentification	
Code source de simulation		-OpenNetInf -NetInf (nilib) -GIN	-Blackadder	-CCN-lite -SCoNet -CCNSim -CCNPL-Sim -Mini-CCNx

Tableau 1: Comparaison entre des principaux projets ICN

- Notre travail se focalise dans le routage des réseaux CCN/NDN, par conséquent nous allons plus approfondir dans le routage de ces derniers

2.4. Routage dans NDN

L'idée générale de NDN est d'adapter les messages envoyés sur Internet à ce qu'ils sont vraiment : le contenu. Au lieu de la restriction aux communications de bout en bout entre les paires d'utilisateurs, NDN permet un échange de messages beaucoup plus flexible et efficace. [8]Dans un NDN, il n'y a nul besoin d'acheminer les adresses source et destination à travers le réseau pour récupérer la donnée. Le format des paquets Intérêts et Data est explicité à la Figure 7.

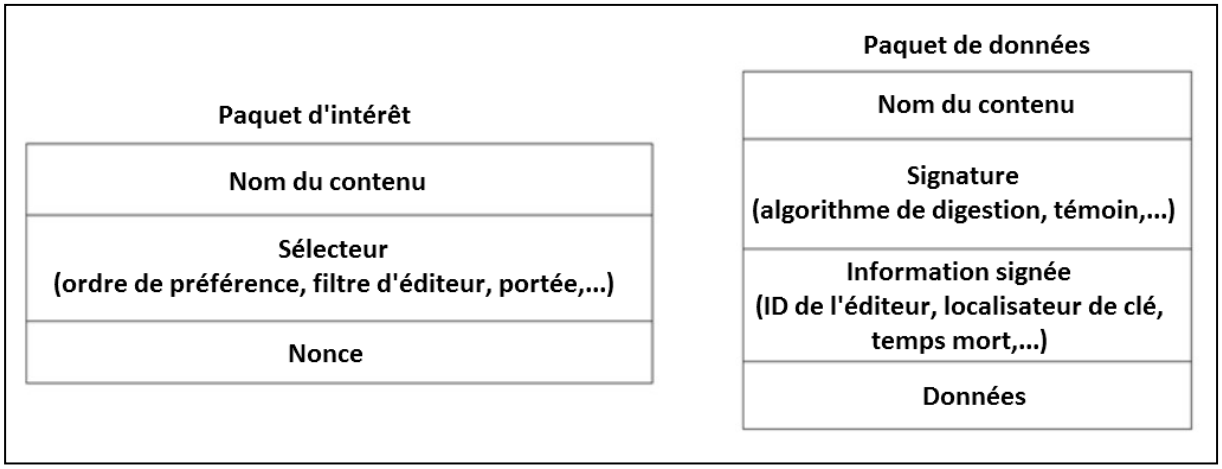

Figure 7: Type de paquet NDN. [7]

Les contenus sont divisés en morceaux (chunks), chaque morceau a typiquement la taille d'un paquet IP. NDN respecte le déroulement logique d'une requête : un utilisateur demande une donnée en émettant des paquets de type "Intérêt" et reçoit en retour des paquets de données de type "Contenu". A chaque paquet Intérêt correspond un seul paquet Contenu et chaque paquet Contenu correspond à un Chunk.

Par rapport à l'Internet actuel, les routeurs sont très différents (voir Figure 8). Ils contiennent essentiellement trois tables : Le stockage de contenu (Content Store), la table (PIT) et la table (FIB)**.** [3]

• ***Content Store (CS) :*** le CS est un cache (ou une mémoire tampon) installé dans les nœuds CCN. Lorsqu'un nœud reçoit des données (message Data), selon les stratégies de caches définies, le nœud peut sauvegarder une copie de ces données dans son CS pour répondre aux demandes ultérieures.

• ***Pending Interest Table (PIT) :*** la table PIT a deux fonctionnalités principales. La première est qu'elle mémorise temporairement des messages d'intérêt « Interest » que le nœud reçoit avant de les transmettre ensuite au nœud suivant. Grâce à cette table, en retour des données, le paquet Data peut suivre les chemins inverses et finalement arriver jusqu'aux clients demandeurs. Le deuxième rôle de PIT est d'éviter de multiple envoi des mêmes messages Intérêt. Lorsque plusieurs messages Intérêt qui demandent un même contenu arrivent sur un nœud, seul le premier est renvoyé pour chercher le contenu, les autres restent dans ce nœud et attendent la réception du contenu. Une fois reçues, les données seront retournées sur chacune des faces présentes dans la PIT.

• ***Forwarding Information Base (FIB) :*** la table FIB de CCN est similaire à celle d'IP. Elle est utilisée pour gérer les informations de transfert des paquets Intérêt vers des sources qui ont les contenus demandés. La table FIB est remplie par des publications de contenus qui sont publiés par des fournisseurs de contenu.

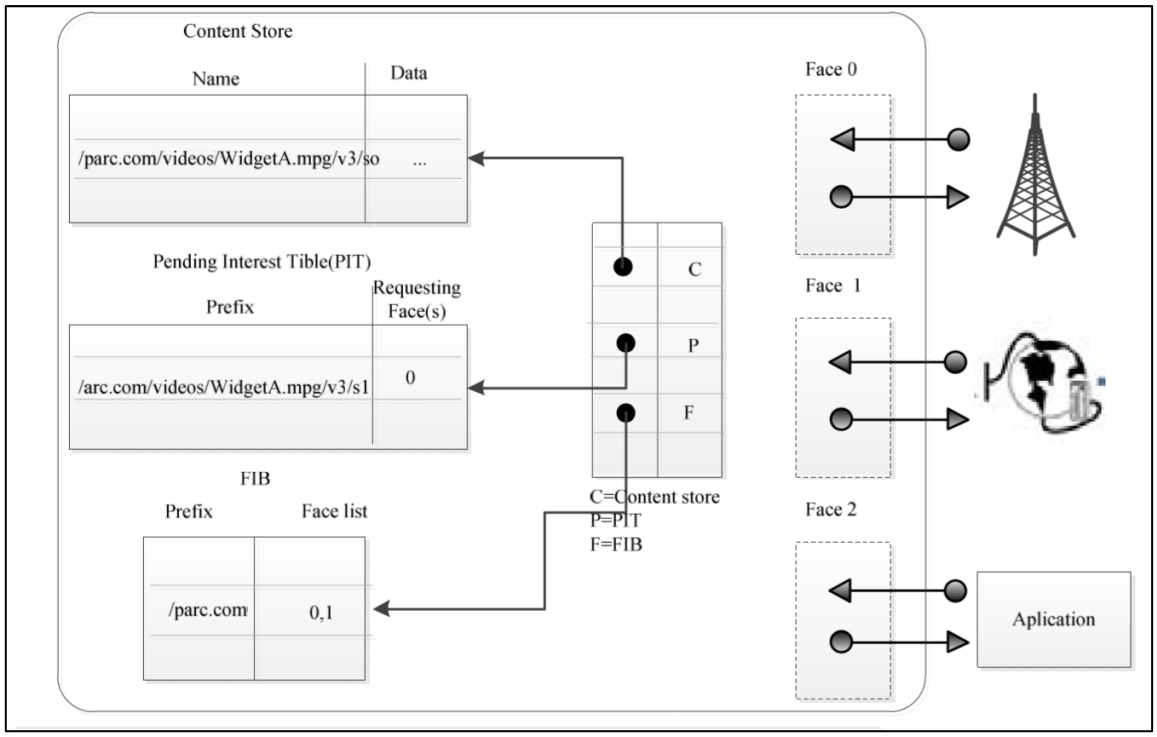

Figure 8: Modèle de transmission NDN. [7]

Lorsqu'un Intérêt est reçu par un nœud, ce dernier vérifie si le chunk demandé existe dans son Content Store. Si c'est le cas, le paquet Contenu sera envoyé à l'interface demandeuse. Sinon le chunk demandé sera recherché dans le PIT. S'il est trouvé, l'interface demandeuse sera rajoutée au PIT. Si les deux bases de données ne fournissent aucune information, on cherchera dans le FIB si une entrée correspond avec le chunk recherché. Alors le paquet Intérêt sera acheminé vers les interfaces conduisant à la donnée. La table PIT sera mise à jour avec une nouvelle entrée pour le chunk en question. [8] A la réception d'un paquet Contenu par un nœud, une recherche est effectuée dans le Content Store. Si une entrée est similaire, alors le paquet reçu est supprimé, car ceci implique que le chunk est déjà livré à toutes les interfaces demandeuses. Sinon la donnée sera recherchée dans le PIT. Si une entrée correspond à la donnée reçue, elle sera acheminée vers les interfaces demandeuses. Le chunk sera typiquement stocké en même temps dans le Content Store.

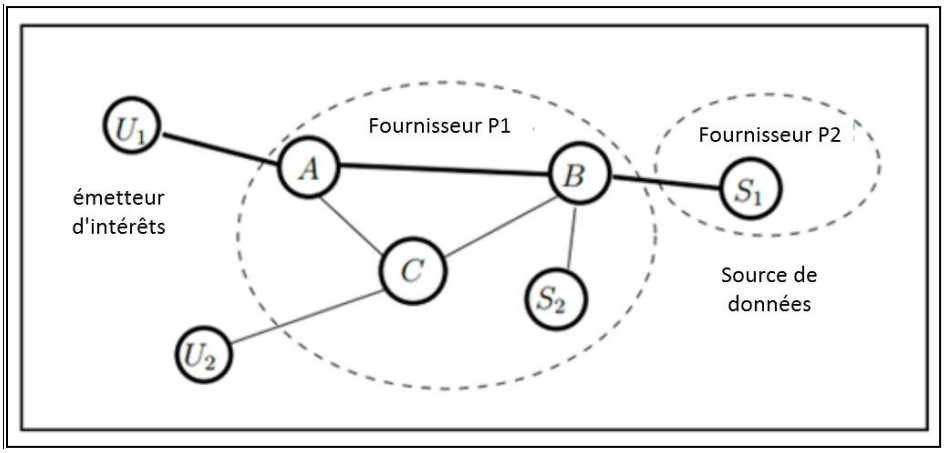

Figure 9: Un segment du réseau NDN reliant un utilisateur U1 à une source S [9]

La Figure 9 représente un segment d'un réseau NDN. Pour récupérer des données du fournisseur P2, l'usager U1 envoie des paquets "Intérêt" pour le contenu demandé au travers des routeurs A et B. Supposant que les Content Stores de A et B ne contiennent pas le document demandé, les paquets Contenu suivent le chemin inverse de S1 vers U1 en passant par B et A.

Dans l'exemple de la figure 10, le nœud A cherche les chunks "vidéo" et "image". Le FIB du nœud A indique que les paquets d'intérêt doivent être acheminés vers l'interface 0 et 1 pour l'image, et vers l'interface 1 pour la vidéo. A la réception de l'intérêt demandant la vidéo par le nœud B, ce dernier ignore l'intérêt reçu car le PIT contient déjà une entrée. Cette entrée est mise à jour. Cependant, quand le nœud B reçoit l intérêt demandant l'image, il l'envoie à l'interface 1 indiquée par le FIB. Le nœud D, par la suite, achemine la donnée vers le nœud A. Cette donnée sera stockée dans tous les Content Stores des nœuds l'ayant reçu ou transité. Le séquencement des paquets est établi grâce aux noms des chunks. Ces derniers sont organisés d'une façon hiérarchique. Ainsi, pour demander un segment il faut indiquer le nom hiérarchique du chunk demandé dans le paquet intérêt. Le FIB détermine l'interface de sortie adéquate grâce à un algorithme "longest prefixe match". [9]

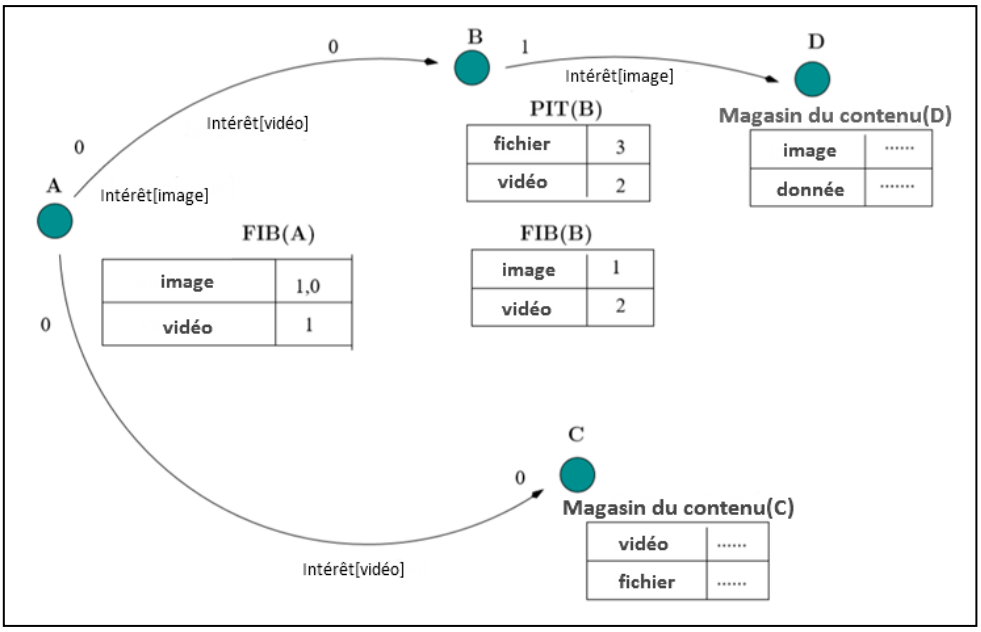

Figure 10: la recherche des données dans NDN. [9]

3. Conclusion

Les réseaux ICNs sont des réseaux centrés contenus qui diffèrent radicalement des réseaux IPs, par leurs architectures et leur principe de fonctionnement. Toutefois, les ICNs ne suivent pas une norme bien définie, plusieurs projets ont été lancés en proposant des protocoles de communication spécifiques. Nous nous sommes approfondis dans le routage de ce dernier car notre travail se focalise principalement dans cette fonction qui nous mène par la suite à étudier ses problèmes de sécurité ce qui fera l'objet du chapitre suivant.

CHAPITRE II : LA SECURITE DANS LES CCNS

1. Introduction

Comme tous les systèmes de réseau informatique les ICNs sont vulnérables à différents types d'attaques qui peuvent les affecter ; ce qui a mené les chercheurs à de nouveaux défis en termes de sécurité et de performances. Ces derniers cherchent de nouvelles solutions de sécurité qui seront intégrées aux architectures ICN et qui garantissent par la suite la confidentialité, la disponibilité et le contrôle d'accès.

Dans ce qui suit, nous allons présenter les problèmes de sécurité de cette nouvelle architecture. Ensuite, nous présentons quelques solutions de sécurité offertes par l'architecture CCN/NDN.

2. Attaques dans les ICNs

2.1. Classification

Après l'apparition d'ICN beaucoup d'efforts ont été faits pour lutter contre les attaques qui peuvent y parvenir. L'ICN doit résoudre de nombreux problèmes de sécurité. D'après Abdallah [5] les attaques ICN sont classées en quatre catégories, comme le montre la Figure 11: attaques de dénomination, de routage, de mise en cache et diverses.

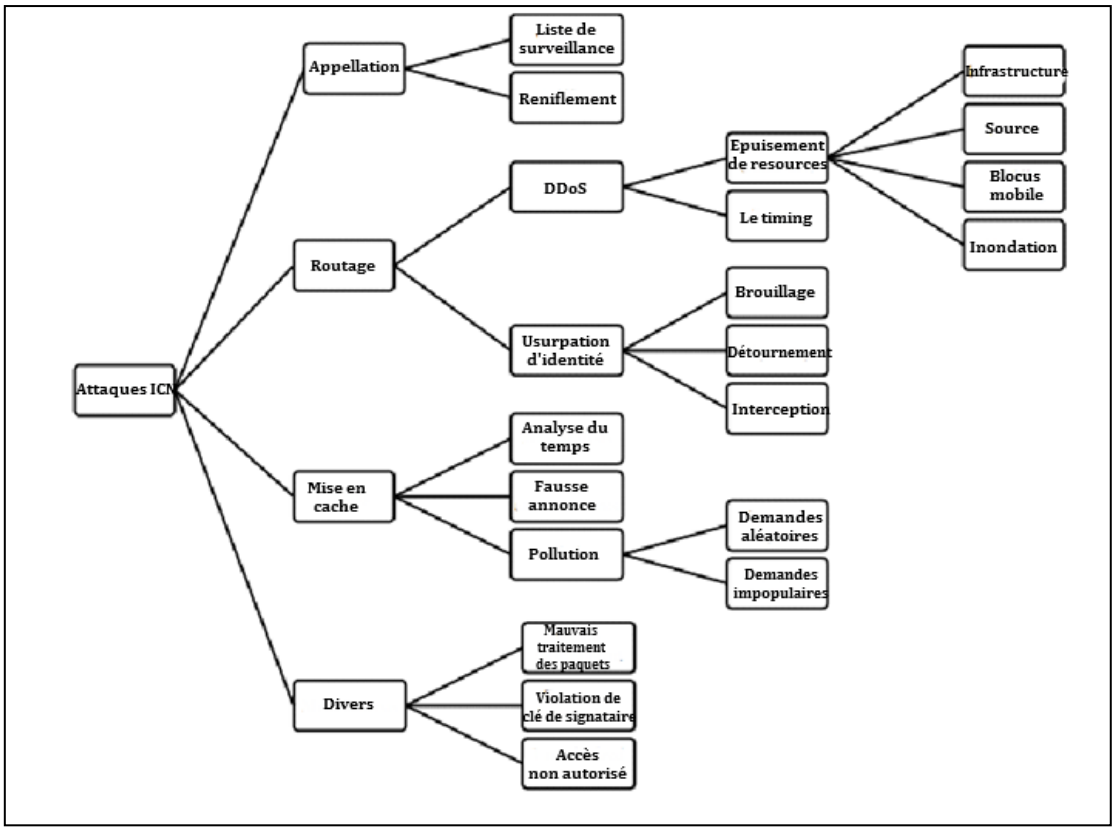

Figure 11: taxonomie des attaques ICN [5]

Cette classification dépend de la cible principale de l'attaquant. Bien que chaque attaque n'apparaisse que sur une catégorie, elle peut également affecter d'autres catégories. Par exemple, les attaques d'inondations et les attaques de demandes impopulaires affectent le routage et la mise en cache ICN. Lors d'une attaque par inondation, l'attaquant a pour objectif principal de surcharger et d'épuiser les ressources de routage, par conséquent, d'affecter le système de mise en cache. Dans les attaques par requête impopulaires, la cible principale de l'attaquant consiste à violer la pertinence du cache et affecter le système de routage.

Les catégories proposées sont brièvement présentées comme suit:

2.1.1. Attaques liées au nommage

Les architectures ICN font face à une plus grande menace en ce qui concerne la confidentialité, car les demandes de contenu sont visibles sur le réseau. De nombreux attaquants tentent de censurer / surveiller l'utilisation d'Internet. Une architecture ICN offre davantage l'accès aux requêtes des utilisateurs, ce qui augmenterait le contrôle des pirates sur le flux d'informations et leur faciliterait grandement le blocage des informations. Dans les attaques de nommage dans ICN, un attaquant tente d'empêcher la diffusion d'un contenu

spécifique en bloquant la diffusion de ce contenu et / ou en détectant qui en fait la demande.

2.1.2. Attaques liées au routage

La diffusion du contenu ICN dépend de la publication et de l'abonnement asynchrones, ce qui ajoute un effort supplémentaire pour assurer la cohérence des états de données distribuées. Certaines attaques, telles que le brouillage et le minutage, visent à faire échec à la cohérence de cet état, ce qui peut entraîner des flux de trafic non souhaités et / ou un déni de service. D'autres attaques, telles que les attaques d'infrastructure et d'inondation, tentent d'épuiser les ressources, telles que la mémoire et la puissance de traitement, utilisées pour prendre en charge, maintenir et échanger des états de contenu. En outre, l'infrastructure dans ICN repose sur l'intégrité et l'exactitude de l'acheminement du contenu et est donc menacée par des injections toxiques de chemins et de noms. C’est sur ce type d’attaque que notre travail se base.

2.1.3. Les attaques liées à la mise en cache

Le cache est l’un des composants importants d’ICN, dans la mesure où les performances de son infrastructure reposent sur une mise en cache pilotée par le récepteur, qui vise à fournir à un utilisateur la copie disponible la plus proche. ICN est donc vulnérable à toutes les opérations polluant ou corrompant le système de mise en cache.

2.1.4. Attaques diverses

Les menaces de cette catégorie visent à dégrader certains services ICN et permettent à un attaquant de faire un accès non autorisé. Ces attaques entraînent une distribution de données insuffisante ou erronée.

Dans ce qui suit, nous décrivons les attaques liées au routage, les scénarios et les impacts de chacune d’entre elles.

2.2. Attaques liées au routage dans le réseau ICN

Les attaques de cette catégorie peuvent être classées en : [5]

2.2.1. Infrastructure

Un attaquant envoie un grand nombre de demandes de contenu disponible / indisponible. Alors que les architectures ICNs tentent de trouver la copie la plus proche à partir du meilleur emplacement disponible, ces demandes empruntent différents chemins vers la source,

provoquant des conditions de surcharge. Si le nombre de ces demandes est significativement élevé, cela entraîne un déni de service. Cette attaque peut être encore amplifiée, car les utilisateurs ordinaires envoient des demandes de retransmission après un délai spécifié. Semblable à l'attaque de piratage, cette menace peut être atténuée car les mécanismes d'acheminement dans ICN tentent d'acheminer vers plusieurs emplacements. Comme illustré à la figure 12, l'attaquant, qui contrôle de nombreux systèmes d'extrémité, envoie un grand nombre de demandes à un ou plusieurs routeurs ICNs afin de remplir la table de routage et d'épuiser les ressources de traitement et de mémoire. En conséquence, les routeurs attaqués transmettent ces demandes aux nœuds voisins, qui les transmettent à leur tour aux nœuds voisins suivants…etc. Si le nombre de demandes non valides est si élevé, toute demande légitime prend un temps de réponse plus long. Par conséquent, si le temps de réponse dépasse le délai d'expiration de la demande, il est alors impossible de répondre à la demande. Ce scénario peut entraîner un déni de service ou au moins des retards importants.

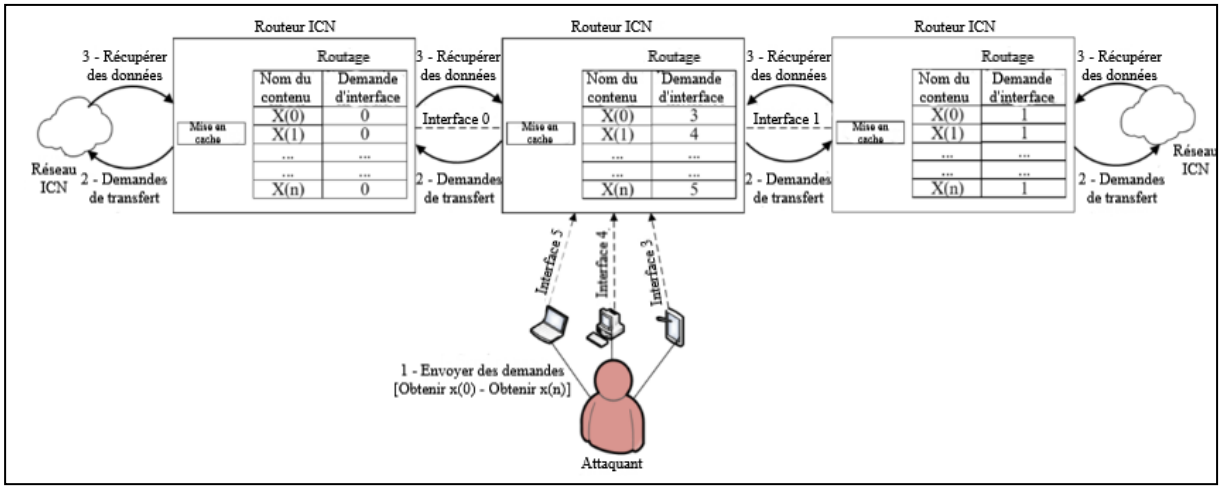

Figure 12: Attaque d'infrastructure. [5]

Scénario d'attaque :

1. Un attaquant, qui contrôle plusieurs systèmes finaux, envoie un grand nombre de requêtes aux routeurs ICN.
2. Les routeurs attaqués transmettent ces requêtes aux routeurs voisins et les envoient à leur tour aux routeurs voisins…etc.
3. ICN commence à extraire ces grandes quantités de données de différents chemins et les renvoie aux emplacements demandés.

2.2.2. Source

En ICN, attaquer une source unique peut également entraîner une surcharge de l'infrastructure de routage. Un attaquant envoie un grand nombre de requêtes à une source de contenu spécifique afin de dégrader ses performances. En conséquence, cette attaque augmente le temps de réponse de la livraison de contenu pour cette source de contenu ou son routeur d'accès [5]. En plus de cet effet, l'attaque peut réduire le taux de retour des données et affecter les demandes de tous les nœuds dans les chemins d'accès aux destinataires. Le scénario d'attaque est similaire au scénario d'attaque d'infrastructure. Cette attaque n'affecte pas seulement la source attaquée, mais également l'ensemble du réseau.

2.2.3. Blocage mobile

Un attaquant mobile peut surcharger une région en traversant des réseaux voisins sur des chemins circulaires tout en envoyant un nombre important de demandes de contenu. L'attaquant a pour objectif de surcharger les routeurs d'accès mobile afin qu'il dépasse le délai d'exécution de l'état qui entraîne un blocage des réseaux disponibles au niveau régional. La retransmission des demandes fait partie de l'aspect de la mobilité dans un environnement ICN qui ajoute une difficulté supplémentaire à la détection de cette attaque. Le scénario d'attaque présenté à la figure 13 est similaire au scénario d'attaque d'infrastructure. La différence est que l'attaquant mobile envoie un grand nombre de requêtes aux réseaux voisins, alors que l'attaquant traverse les réseaux de manière circulaire et continue.

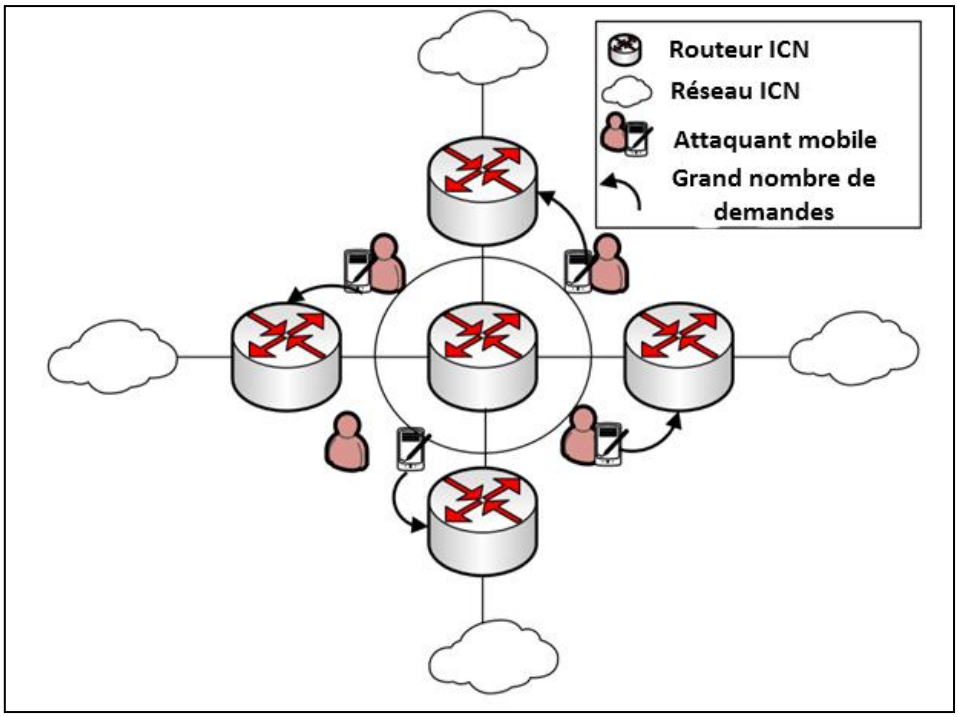

Figure 13: Attaque par blocage mobile [5]

Un attaquant mobile envoie un grand nombre de requêtes alors qu'il / elle traverse des réseaux voisins ICN.

2.2.4. Minutage (Timing)

Cela fait référence à l'augmentation du délai d'expiration de la demande pour certains nœuds ICN afin de violer la cohérence entre la publication asynchrone ICN et le processus de souscription. Un attaquant envoie un grand nombre de demandes afin de dégrader les performances de certains routeurs, de sorte que le routage des demandes et le transfert des données présentent des délais plus longs. Le scénario d'attaque est également similaire au scénario d'attaque d'infrastructure. La différence est que l'attaquant envoie un grand nombre de demandes via un ou plusieurs itinéraires afin d'augmenter le délai d'expiration du délai imparti aux demandes des utilisateurs légitimes.

2.2.5. Brouillage (Jamming)

Un nœud sur un lien partagé envoie un grand nombre de demandes de contenu inutiles et malveillants. L'attaquant qui se fait passer pour un abonné de confiance envoie les requêtes malveillantes afin de perturber le flux d'informations dans le système. Le réseau ICN répond et le contenu est envoyé à la destination sans récepteur. Ce scénario d'attaque est similaire au

scénario d'attaque d'infrastructure. La différence, présentée à la figure 14, réside dans le fait que l'attaquant envoie des requêtes à un nœud partagé, qui les transmet aux nœuds voisins.

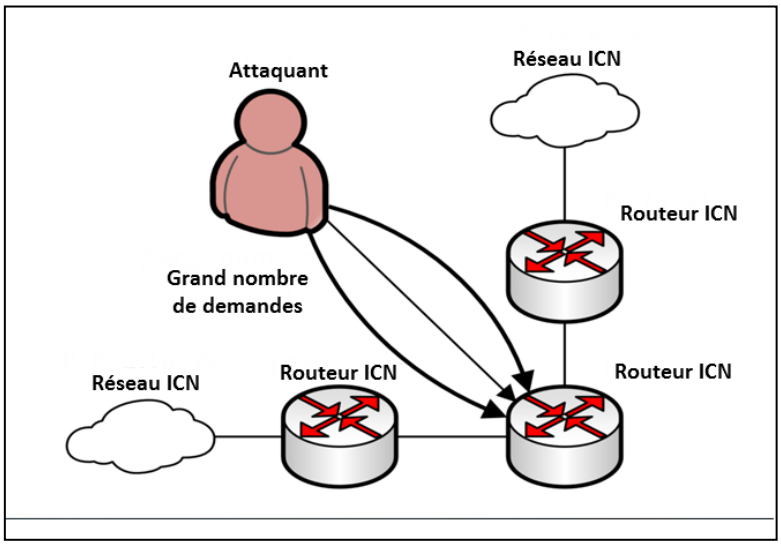

Figure 14: Attaque par brouillage [5]

Un attaquant envoie un grand nombre de requêtes à un nœud partagé.

2.2.6. Détournement (Hijacking)

Contrairement aux architectures centrées sur l'hôte, tout nœud d'ICN peut mettre en cache et publier / souscrire du contenu. Un attaquant qui se fait passer pour un éditeur de confiance peut annoncer des itinéraires invalides pour tout contenu. Les demandes de contenu émanant d'utilisateurs situés à proximité de l'attaquant sont dirigées vers ces itinéraires non valides. Par conséquent, ces demandes resteront sans réponse, ce qui conduira à un déni de service. L'effet de cette attaque peut être exacerbé si l'attaquant a la capacité de détourner à grande échelle des itinéraires non valides. L'effet de cette attaque est atténué car les mécanismes de routage dans ICN tentent de router vers plusieurs emplacements. Comme illustré à la figure 15, l'attaquant annonce des itinéraires non valides pour certains contenus afin d'attirer les requêtes des utilisateurs. Lorsque des utilisateurs légitimes envoient des demandes pour l'une de ces routes malveillantes, les nœuds ICN les transmettent aux nœuds malveillants. Par conséquent, l'utilisateur légitime ne reçoit pas de réponse.

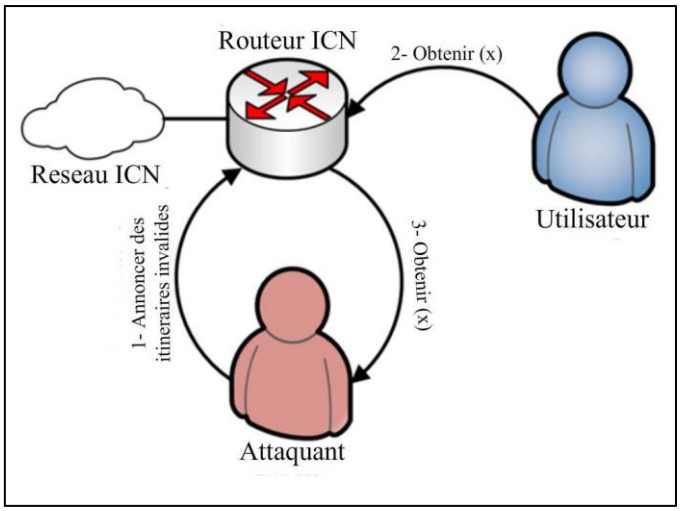

Figure 15: Attaque de détournement [5]

Scénario d'attaque :

1. Un attaquant annonce les itinéraires invalides pour certains contenus, y compris (x).
2. Un utilisateur demande un contenu ICN nommé (x).
3. Le routeur ICN redirige les requêtes de l'utilisateur vers les routes malveillantes de l'attaquant et, par conséquent, l'utilisateur n'obtient aucune réponse.

2.2.7. Interception

Cette attaque est similaire à l'attaque habituelle «homme au milieu». Contrairement à une attaque de piratage, un attaquant qui se fait passer pour un éditeur de confiance annonce les itinéraires non valides, tout en conservant un enregistrement des itinéraires valides vers le contenu. Les demandes de contenu peuvent ensuite être capturées et envoyées au bon endroit. Bien que le destinataire obtienne le contenu normalement, l'attaquant acquiert la connaissance du contenu demandé. Comme le montre la figure 16, l'attaquant annonce des itinéraires non valides pour certains contenus afin d'attirer les requêtes de l'utilisateur. Lorsque des utilisateurs légitimes envoient des demandes pour l'un des itinéraires malveillants, les nœuds ICN les transmettent au nœud malveillant de l'attaquant. L'attaquant enregistre qui a demandé ce contenu, puis le transmet pour obtenir les données réelles. Lorsque les données réelles parviennent au nœud de l'attaquant, l'attaquant les renvoie au serveur demandé. Nœud ICN, qui le transmet à son tour à l'utilisateur légitime. Pour l'utilisateur, le scénario semble normal, mais l'attaquant viole en réalité la vie privée de l'utilisateur. [5]

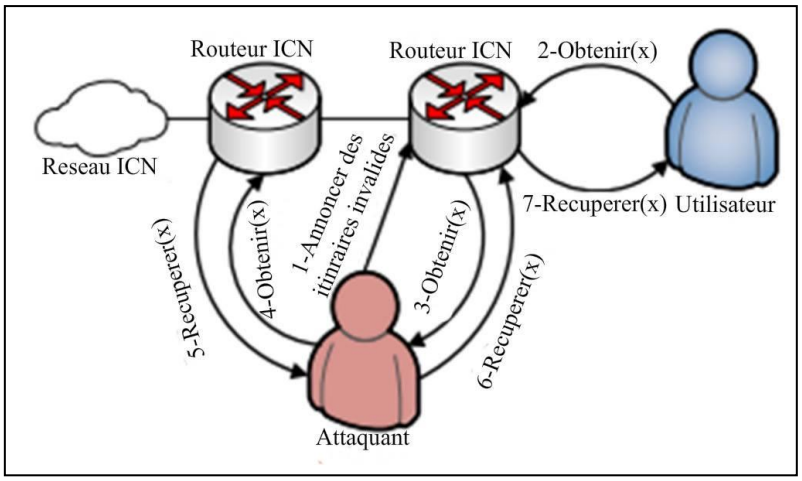

Figure 16: Attaque d'interception. [5]

Scénario d'attaque :

1. Un attaquant annonce les itinéraires invalides pour certains contenus, y compris (x).
2. Un utilisateur demande un contenu ICN nommé (x).
3. Le routeur ICN redirige la demande de l'utilisateur vers les routes malveillantes de l'attaquant.
4. L'attaquant transmet la demande pour obtenir le contenu réel.
5. L'attaquant récupère le contenu (x).
6. L'attaquant transmet le contenu à l'utilisateur demandé.
7. L'utilisateur récupère le contenu (x).

3. Hypothèse

Dans les NDNs la notion de spoofing, appelée plus communément usurpation d'identité, reste toujours aussi fidèle à son homologue IP, mais avec quelques différences mineures dans son fonctionnement et de sa mise en place, ce type d'attaque cause des dégâts non négligeables entre l'utilisateur légitime et le service qui fournis ou transmet les données.

Au cours ce chapitre, nous allons présenter une des attaques liées au spoofing qui est : l'interception, ainsi nous allons aborder son déroulement, les failles qu'elles exploitent et leur gravité sur le réseau de façon générale, pour aux finales proposer des solutions pour contrer ces types d'attaques

4. L'attaque d'interception

Les attaques liées au routage sont classées en deux catégories [13] : attaques orientées FIB et attaques orientées PIT (Figure 17) dont les cibles sont deux composantes liées au routage dans un routeur NDN. En NDN, le réseau se charge de fournir le contenu correct à l'utilisateur à partir d'un emplacement optimal. Afin d'atteindre cet objectif, la cohérence entre les bases d'informations de routage distribué doit être assurée. Essayer de corrompre cette unité est la cible principale des attaques liées au routage.

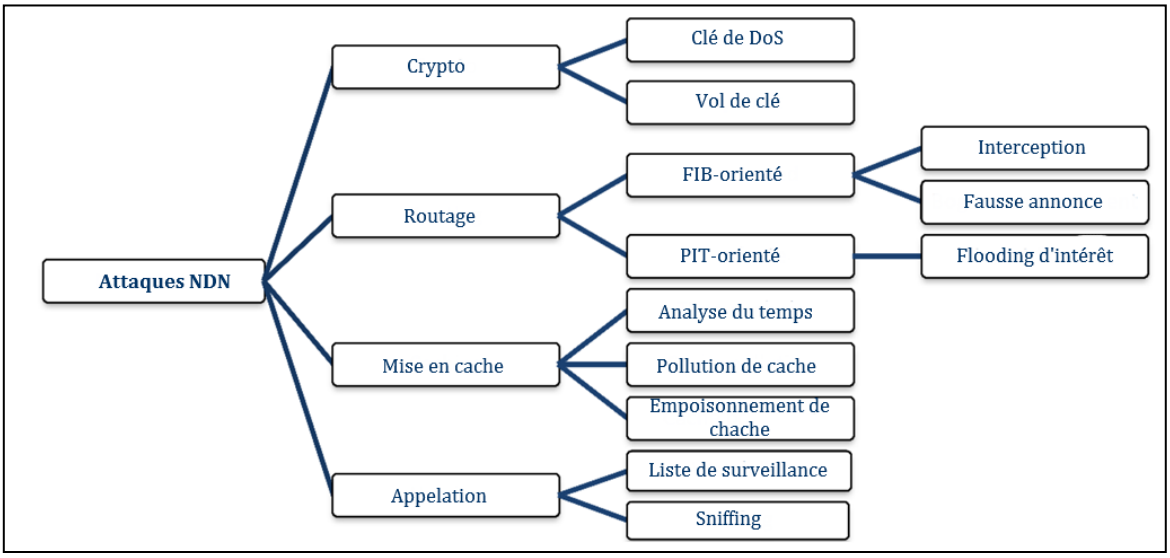

Figure 17: taxonomie des attaques de sécurité dans NDN [12]

Dans le scénario d'interception (Figure 18), les routeurs NDN reçoivent des informations de routage erronées et transmettent les demandes de renseignements vers des itinéraires malveillants. L'architecture NDN ayant pour objectif d'améliorer les performances de livraison de contenu, il est tout à fait possible pour les entités d'annoncer leurs contenus et leurs copies mises en cache à d'autres nœuds du réseau. Tirant parti de cette fonctionnalité, un attaquant peut exécuter l'attaque de piratage en annonçant des informations de routage falsifiées, par exemple en demandant à d'autres nœuds de lui transmettre les demandes de contenu. Ensuite, l'attaquant peut surveiller les requêtes des utilisateurs proches et même les bloquer pour provoquer un effet de Déni De Service.

L'attaquant peut rendre son comportement encore plus sophistiqué en interceptant, c'est-à-dire en transmettant les demandes des utilisateurs aux emplacements appropriés après les avoir capturées, mais en modifiant les données renvoyées sous le nom de contenu demandé.

Tout semble être normal pour les utilisateurs, mais ils ne réalisent pas que leurs demandes ont été enregistrées et ils pourraient obtenir un faux contenu.

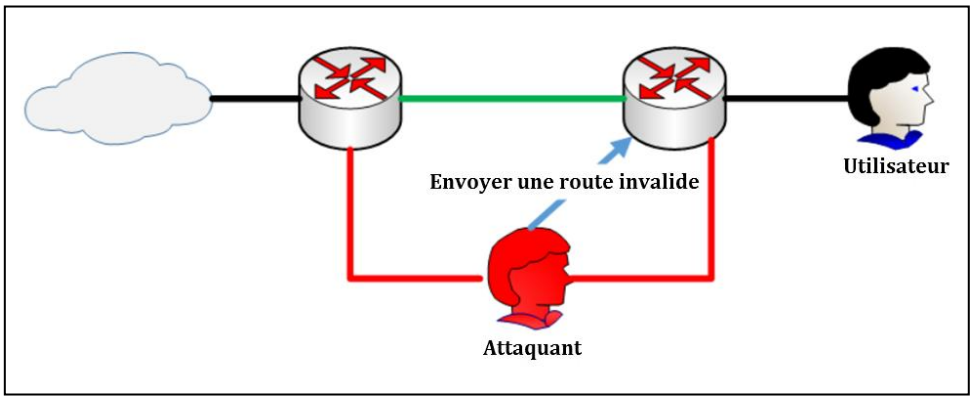

Figure 18: Scénario d'interception

4.1. Description du problème

Le FIB est une structure fondamentale utilisée pour la transmission des messages d'intérêt vers le contenu désiré. Elle peut être modifiée lors des mises à jour faites statiquement ou dynamiquement. Par conséquent la mise à jour de la table FIB peut mener à des attaques dans le réseau NDN car à cause de certaines failles de ce dernier un utilisateur illégitime pourra espionner et censurer les utilisateurs afin de se renseigner des personnes, de leurs demandes et des contenus qu'elles reçoivent ce qui mène à la divulgation de l'anonymat et de la vie privée des autres.

Dans le tableau 1 nous identifions trois attributs susceptibles d'accroître l'impact d'attaque d'interception dans le CCN et le degré que cette dernière peut en dépendre pour qu'elle soit exécutée. Ces attributs sont les suivants: [5]

Caractéristiques	Nommage indépendant de l'emplacement	Dans la mise en cache réseau	Publication / abonnement omniprésente
Dégréé de dépendance	Totalement	Partiellement	Totalement

Tableau 2: Dépendance des défauts de CCN face à l'attaque interception [5]

Voici une description des caractéristiques :

4.1.1. Nommage indépendant de l'emplacement

Cet attribut permet la récupération de contenu à partir de plusieurs emplacements inconnus ou non approuvés. ICN a besoin d'un système de nommage sécurisé pour nommer le contenu indépendamment de son emplacement et de sa représentation.

4.1.2. Mise en cache dans le réseau

La mise en cache est l'une des principales caractéristiques des architectures CCN. Tout nœud du réseau peut mettre en cache tout élément qui le traverse. Le contenu peut être délivré à partir du cache le plus proche qui contient le contenu au lieu d'aller sur le serveur d'hébergement.

4.1.3. Publication / abonnement omniprésente

Tout utilisateur peut accéder au réseau CCN depuis n'importe quel endroit et agir en tant que fournisseur de contenu ou consommateur de contenu. Certains utilisateurs peuvent envoyer des contenus ou des demandes indésirables.

4.2. Impacts

4.2.1. Infiltration de chemin

Dans CCN, les copies de contenu sont généralement distribuées dans de nombreux emplacements non approuvés. Il est donc difficile d'authentifier les origines valides pour le contenu.

L'interception est la principale source d'infiltration de chemins dans ICN, car les attaquants peuvent annoncer des routes non valides et les revendiquer comme des routes fiables. [5]

4.2.2. Intimité

La violation de la confidentialité dans l'attaque d'interception donne à l'attaquant un accès non autorisé aux demandes de l'utilisateur, en particulier lorsque l'attaquant est topologiquement proche ou sur la route vers l'utilisateur. [5]

4.3. Scénario d'attaque

Dans les réseaux IPs l'attaque d'interception se fait à l'aide de différents outils comme kali linux…etc. Par contre dans les réseaux ICNs aucun outil n'a encore été développé, par conséquent on suppose que l'attaquant est un administrateur malicieux qui va orienter les

routeurs vers une fausse destination qui est celle d'un attaquant complice en accédant aux fichiers **routeur_ccn.cfg** et en modifiant la table **eFwdRulesMode.**

4.4. Les solutions possibles

Pour protéger la vie privée des utilisateurs plusieurs solutions ont été proposées. Citons quelques-unes :

- Bien entendu, les producteurs de contenu pourraient simplement générer une nouvelle paire de clés pour signer chaque paquet de contenu. Cela serait peu pratique, car les coûts élevés de la production et de la distribution des clés rendraient difficile l'authentification du contenu par les consommateurs.
- La signature de contenu d'origine pourrait être remplacée par celle générée par un AR. Cependant, cela empêcherait la vérification de contenu de bout en bout et romprait ainsi le modèle de confiance NDN.

5. Les impacts liés aux attaques de routage

5.1. Déni de service

Les dénis de service peuvent survenir en raison de nombreuses attaques de cette catégorie, telles que l'envoi de nombreuses demandes de contenu indisponible ou à une source unique, le bloquage mobile, l'inondation, le piratage et le minutage. Par conséquent, les temporisateurs intermédiaires suppriment les demandes dont les délais sont expirés, ce qui peut entraîner des dénis de service ou au moins des retards importants.

5.2. Epuisement des ressources

Il existe de nombreuses sources d'épuisement des ressources dans l'infrastructure ICN qui résultent d'une utilisation abusive ou d'un trafic incontrôlé, telles que l'envoi d'un grand nombre de demandes et les attaques par inondation.

5.3. Infiltration du chemin

Dans ICN, les copies de contenu sont généralement distribuées à de nombreux emplacements non fiables. Il est donc difficile d'authentifier des origines valides pour le contenu.

Les détournements et les interceptions sont les principales sources d'infiltration de chemins

dans ICN, car les attaquants peuvent annoncer des routes non valides et les revendiquer comme des routes fiables.

5.4. Intimité

La violation de la confidentialité dans l'attaque par interception donne à l'attaquant un accès non autorisé aux demandes de l'utilisateur, en particulier lorsque l'attaquant est topologiquement proche ou sur la route qui le relie à l'utilisateur.

6. Sécurité et confidentialité du NDN

Dans cette section, nous résumons les solutions de sécurité offertes par les réseaux NDNs en termes de sécurité de la vie privée, et l'intégrité du contenu. [8]

6.1. L'intégrité vérifiable

Dans NDN/CCN le producteur de contenu doit fournir une signature pour chaque paquet de contenu, ce qui garantit aux consommateurs l'intégrité des données reçues, cet aspect positif pourra diminuer d'une manière remarquable les attaques de spoofing qu'on voit déjà dans les réseaux IPs.

6.2. L'absence des adresses

Comme nous avons déjà parlé des principes fondamentaux des réseaux ICNs, l'acheminement des paquets dans NDNs se fait à base de noms que d'adresses. Les intérêts dans CCNs contiennent uniquement le nom du contenu demandé, mais pas celui qui l'a demandé. Seul le premier routeur d'acheminement connaît l'interface à partir de laquelle le contenu a été demandé. Tous les autres routeurs connaissent uniquement le routeur précédent sur le chemin de transfert. Lorsque le fournisseur de contenu renvoie le contenu, son message comprend également le nom du contenu, sa signature, l'ID de l'éditeur et des informations sur l'endroit où récupérer la clé publique de l'éditeur. Par conséquent, il n'est pas nécessaire d'adresser des adresses ainsi que l'éditeur puisse être déduit de l'ID et de la clé. Du point de vue de la vie privée, ce design améliore l'anonymat car la source d'un intérêt est inconnue ou au moins difficile à découvrir. Cet aspect permet d'éviter toutes les attaques qui exigent à l'attaquant l'envoie des messages à la victime, de plus aucun message ne pourra être arrivé à l'hôte sans que ce dernier ne le demande.

6.3. Protection contre le déni de service

Outre la protection contre les attaques directes sur les hôtes, CCN offre même une protection contre les attaques de déni de service (DoS) sur les fournisseurs de contenu. Supposons qu'un attaquant lance une attaque DoS sur un éditeur en envoyant beaucoup d'intérêts. Ceux-ci sont simplement collectés au premier routeur et un seul intérêt est transmis. Même si l'attaquant contrôle un grand botnet qui est distribué sur de nombreux endroits qui ne partagent pas les routeurs sur le chemin de routage vers l'éditeur, l'agrégation des intérêts atténue la plupart des attaques.

7. Conclusion

Certes que les réseaux ICNs offrent un système de sécurité important notamment dans le routage par rapport au réseau actuel en particulier les CCNs/NDNs grâce à leur fonctionnement tel que le nommage indépendant de l'emplacement, l'absence d'adressage…etc. Cela n'empêche pas d'y avoir des faiblesses qui peuvent survenir à de différents types d'attaques, tel que le DDOS et le Spoofing et qui ont un impacte sur la confidentialité et l'intégrité des données…etc.

Nous avons abordé dans ce chapitre l'attaque d'interception qui est notre objet d'étude par la description de sa catégorie, qui est FIB-ORIENTED-ATTACK, l'explication de son scénario d'attaque ainsi que les failles à exploiter pour une telle attaque, en passant aux impacts qui peuvent y survenir telle que la violation de la vie privé, enfin nous avons cité quelques solutions proposées mais qui ont un impacts sur le coût et les performances.

CHAPITRE III :

CONCEPTION ET IMPLEMENTATION

1. Introduction

Malgré les différentes solutions proposées pour contrer les attaques d'interception les problèmes d'insécurité restent toujours présents car un intérêt chiffré produit toujours le même résultat, de même les routeurs sont des ressources publiques ce qui aide l'adversaire à décrypter les intérêts précédemment observés, d'observer la sortie correspondante et corréler les intérêts entrants/ sortants. D'autres solutions ont été proposées pour réduire ce risque d'insécurité telle que l'utilisation des canaux chiffrés [8] entre les parties de communication et le mixage (pour le trafic à tolérance de retard) mais qui ont un impact significatif sur les coûts de calcul et la latence.

Pour atténuer les attaques d'interception tout en assurant la confidentialité et l'intégrité sur les NDNs, nous allons proposer dans ce chapitre une approche asymétrique qui se base sur la sécurisation des paquets de contenu à l'aide d'une encapsulation chiffrée de ces derniers. Nous allons faire aussi une simulation sur les CCNs à travers le simulateur « OMNET++ » et le package CCN-lite que nous allons détailler par la suite et enfin terminer ce travail par un scénario d'attaque d'interception avec la solution proposée.

2. Protocole de confidentialité asymétrique

Nous présentons maintenant une technique de confidentialité asymétrique que nous avons implémentée dans le but de renforcer la confidentialité et d'assurer l'intégrité du trafic NDN et afin de permettre un routage efficace des paquets d'intérêt.

Selon notre proposition de confidentialité asymétrique notre objectif est de :

- Assurer la confidentialité du contenu à l'aide d'une encapsulation chiffré de ce dernier entre le consommateur et le producteur.
- Garantir l'authenticité, l'intégrité, et aussi la non-répudiation du contenu à l'aide d'une signature numérique des contenus.

2.1. Encapsulation chiffré du contenu

Le chiffrement est un élément crucial dans la protection des données en les permettant à être inintelligibles pour toutes personnes non autorisée à y accéder. Il existe deux types de chiffrement : [19]

- **Le chiffrement symétrique :** qui est définis de l'utilisation de la même clé pré-partagée (Pre-Shared Key) pour le chiffrement et le déchiffrement des données (voir la figure 19).

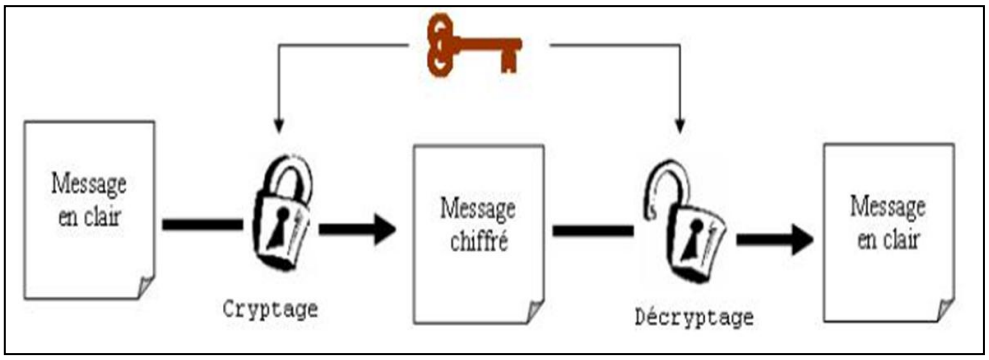

Figure 19: chiffrement symétrique à clé pré-partagée [19]

- **Le chiffrement asymétrique :** contrairement au chiffrement symétrique, le chiffrement asymétrique utilise une paire de clés, une qui est utilisée pour chiffrer le message connu sous le nom de clé publique, et une clé privée pour le déchiffrement du message (voir la figure 20).

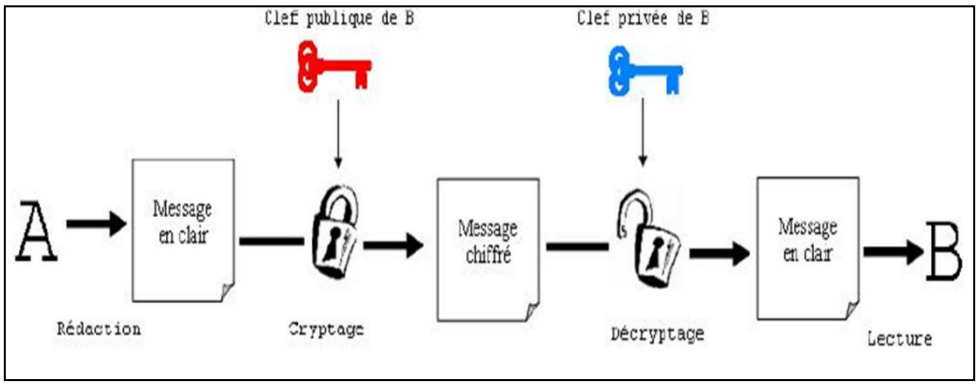

Figure 20: chiffrement asymétrique à clé privée [19]

Parmi ces deux parfums de chiffrement nous avons utilisé le chiffrement asymétrique car par cette méthode nous n'aurons pas le problème de la transmission sécurisée des clés qu'on trouve dans le chiffrement symétrique, de plus il va nous permettre d'établir la signature.

2.1.1. Application de RSA dans l'encapsulation chiffré

L'acronyme du chiffrement RSA vient des initiations de ses trois fondateurs Ronald Rivest, Adi Shamir et Leonard Adleman qu'ils ont publié en 1978 dans Méthode d'obtention de signatures numériques et de systèmes cryptographiques à clé publique.

Cette méthode utilise une paire de clés (des nombres entiers) composée d'une **clé publique**

pour chiffrer et d'une **clé privée** pour déchiffrer des données confidentielles. Les deux clés sont créées par le destinataire du message chiffré (voir la figure 21), qui souhaite que lui soient envoyées des données confidentielles. Le destinataire rend la clé publique accessible. Cette clé est utilisée par ses correspondants (les émetteurs) pour chiffrer les données qui lui sont envoyées. La clé privée est quant à elle réservée par le destinataire, et lui permet de déchiffrer ces données.

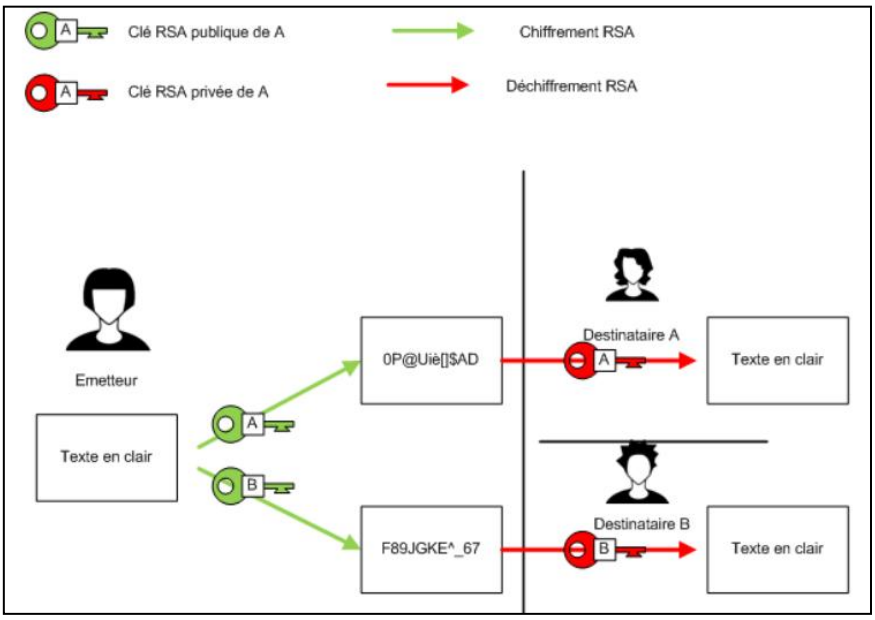

Figure 21: procédure de chiffrement selon RSA [19]

2.1.1.1. Création des clés

L'étape de création des clés est à la charge du serveur. Elle n'intervient pas à chaque chiffrement car les clés peuvent être réutilisées. Le renouvellement des clés n'intervient que si la clé privée est compromise, ou par précaution au bout d'un certain temps (qui peut se compter en milliers d'années).

1. Choisir ***p*** et ***q***, deux nombres premiers distincts.
2. calculer leur produit $n = pq$, appelé ***module de chiffrement.***
3. calculer $\varphi(n) = (p - 1)(q - 1)$ (c'est la valeur de l'indicatrice d'Euler en n) .
4. choisir un entier naturel ***e*** premier avec $\varphi(n)$ et strictement inférieur à $\varphi(n)$, appelé ***exposant de chiffrement***.

5. calculer l'entier naturel ***d***, inverse de ***e*** modulo **φ(*n*),** et strictement inférieur à **φ(*n*)**, appelé ***exposant de déchiffrement*** ; ***d*** peut se calculer efficacement par l'algorithme d'Euclide étendu.

Comme ***e*** est premier avec **φ(*n*)**, d'après le théorème de Bachet-Bézout il existe deux entiers ***d*** et ***k*** tels que $ed = 1 + k\varphi(n)$, c'est-à-dire que $ed \equiv 1 \pmod{\varphi(n)}$: ***e*** est bien inversible modulo **φ(*n*).**

Le couple **(*n*, *e*)** ou **(*e*, *n*)** est la ***clé publique*** du chiffrement, alors que sa ***clé privée*** est le nombre ***d***, sachant que l'opération de déchiffrement ne demande que la clé privée ***d*** et l'entier ***n***, connu par la clé publique (la clé privée est parfois aussi définie comme le couple **(*d*, *n*)** ou le triplet **(*p*, *q*, *d*)).**

2.1.1.2. Chiffrement du message

Si *M* est un entier naturel strictement inférieur à *n* représentant un message, alors le message chiffré est représenté par :

- $C \equiv M^e (mod\ n)$

L'entier naturel ***C*** étant choisi strictement inférieur à ***n***.

2.1.1.3. Déchiffrement du message

Pour déchiffrer ***C***, on utilise ***d***, l'inverse de *e* modulo **(*p* − 1) (*q* − 1)**, et l'on retrouve le message clair ***M*** par :

- $M \equiv C^d (mod\ n)$

Dans notre solution, le chiffrement des paquets de contenu se fait par le serveur en utilisant la clé publique qui sera donnée par le client, il sera encapsulé par la suite et envoyé. A la réception du paquet chiffré, le client va le déchiffrer à l'aide de la clé privée correspondante**.**

2.2. Signature numérique des paquets de contenu

Une signature numérique est un schéma mathématique permettant de vérifier l'authenticité de messages ou de documents numériques. Une signature numérique valide, dans laquelle les

conditions préalables sont remplies, donne au destinataire une très bonne raison de croire que le message a été créé par un expéditeur connu et que le message n'a pas été modifié en transit.

Plusieurs méthodes de signature des documents ont été proposées pour garantir l'intégrité des données et l'authenticité de l'émetteur dans le réseau IP, nous avons choisi une approche parmi les plus répandu qui est celle du PKI (Public Key Infrastructure).

L'établissement d'une signature PKI se résume en trois principales étapes, qui sont les suivantes : [20]

2.2.1. Le hachage du document

Pour assurer l'intégrité des données nous devons mettre en évidence une fonction permettant d'assurer que l'information n'a pas subi de modification. Une fonction de hachage est typiquement utilisée pour vérifier l'intégrité de données, c'est-à-dire qu'à partir d'une donnée fournie en entrée de calculer une empreinte numérique servant à identifier rapidement la donnée initiale, cette empreinte est aussi appelée somme de contrôle ou condensé. La figure 22 illustre comment utiliser une fonction de hashage pour vérifier l'intégrité d'un document numérique.

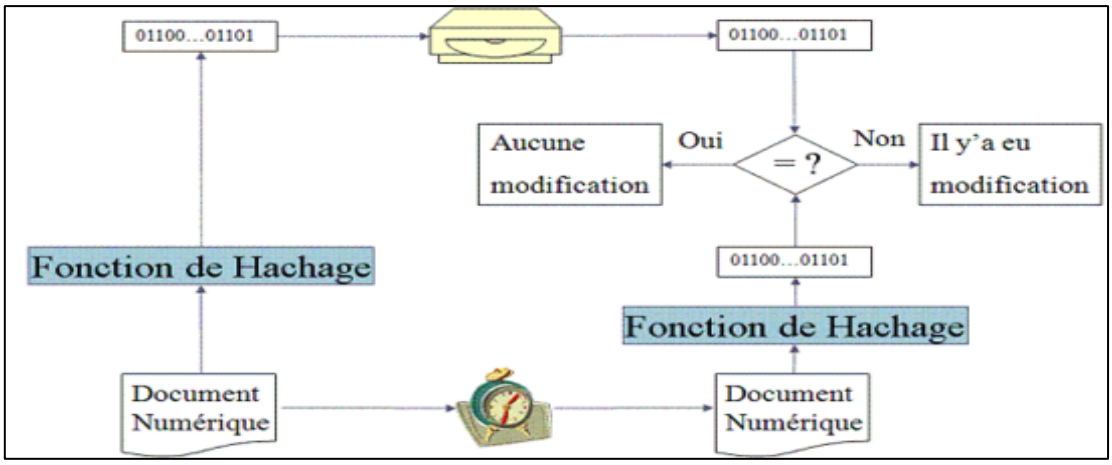

Figure 22: Procédure de vérification de l'intégrité d'un document à l'aide d'une fonction de hashage [20]

Initialement le condensé (code hash) du document numérique est calculé et stocké dans un endroit sûr. Ultérieurement ce code est recalculé et comparé à celui qui a été stocké.

Si les deux valeurs sont égales alors le document n'a pas été modifié. Sinon, le document a subit une modification.

2.2.2. Signature du document hash

Dans notre solution, nous avons fait le hashage à l'aide d'une fonction de hashage MD5. Après l'obtention du hash de contenu, nous allons le signer avec le chiffrement RSA précédemment expliqué en utilisant cette fois la clé privée.

2.2.3. Vérification de la signature

A la réception du contenu qui aurait été initialement chiffré et signé, le consommateur à son tour va le déchiffrer et vérifier la validité de la signature du producteur comme suit :

1. Il va déchiffrer la signature avec la clé publique de RSA, et va obtenir par conséquent le code hash
2. Il déchiffrera par la suite le contenu avec la clé privée.
3. Après avoir obtenu le contenu en clair, il va appliquer un hashage MD5 sur ce dernier.
4. Enfin, il va comparer les deux hashs, celui qui a été reçu avec celui qui a été calculé et voir s'ils sont identiques ça veut dire que l'intégrité du document est valide de plus le contenu a été transmis par le serveur.

3. Simulateur OMNET++

Dans ce projet, nous allons réaliser nos expérimentations à l'aide de OMNET++ qui est un simulateur à évènements discrets orienté objet, basé sur C++. Il a été conçu pour simuler les systèmes réseaux de communication, les systèmes multi processeurs, et d'autres systèmes distribués. OMNET++ est un projet open source dont le développement a commencé en 1992 par Andras Vargas à l'université de Budapest. Actuellement, ce simulateur est utilisé par des dizaines d'universités pour la validation de nouveaux matériels et logiciels, ainsi que pour l'analyse de performance et l'évaluation de protocoles de communication tel que : [17]

- La modélisation des protocoles de communications.
- La modélisation des réseaux filaires et sans fils.
- La modélisation des systèmes répartis.
- Les architectures Hardware.

- En général, il peut être utilisé pour n'importe quel système à évènements discrets pouvant être modélisé selon des entités communiquant par envoient de messages.

L'avantage de OMNET ++ est sa facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés. Nous introduisons dans la suite l'architecture du simulateur OMNET++ et les librairies Mobility Framework et INET [7]

3.1. Architecture d'OMNET++

L'architecture d'OMNET++ est hiérarchique composée de modules. Un module peut être soit simple ou bien composé. Les feuilles de cette architecture sont les modules simples qui représentent les classes C++. Pour chaque module simple correspond un fichier **.cc** et un fichier **.h**. Un module composé est composé de simples modules ou d'autres modules composés connectés entre eux. Les paramètres, les sous modules et les ports de chaque module sont spécifiés dans un fichier **.ned**.

La communication entre les différents modules se fait à travers les échanges de messages. Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service. Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module.

La conception d'un réseau se fait dans un fichier **.ned** et les différents paramètres de chaque module sont spécifiés dans un fichier **.ini**. OMNET++ génère à la fin de chaque simulation

deux nouveaux fichiers **omnet.vec** et **omnet.sca** qui permettent de tracer les courbes et calculer des statistiques.

4. CCN-lite

Le simulateur OMNeT++ n'est pas spécialisé pour les réseaux centrés Contenu (CCN). Pour cela, il existe plusieurs extensions, plateformes et simulateurs basés sur OMNET++ qui essayent d'introduire la notion du CCN dans OMNeT++. La plus utilisée (bien documentée) et la plus facile à manipuler est "CCN-Lite".

4.1. Présentation de CCN-lite

CCN-lite est une implémentation réduite et légère, fonctionnellement compréhensible. Elle couvre :

- des protocoles CCNx et NDN [10].
- Le protocole de réseau centré sur le contenu de PARC.
- Le projet Named-Data Networking (NDN).
- Le propjet Naming-Function Networking.
- Un encodage expérimental et compact pour les environnements IoT.

La motivation initiale pour créer CCN-lite en 2011 était que le logiciel de routage CCNx de PARC (Palo Alto Research Center) est devenu énorme. CCN-lite fournit une alternative maigre à des fins éducatives pour ceux qui veulent un logiciel simple pour leurs propres expérimentations ou développements et qui n'ont pas besoin de toutes les fonctionnalités du CCN. Il nous offre :

- Un noyau CCNx écrit en langage C (1000-2000 lignes).
- Un Support de multiple plate-forme (Espace utilisateur Linux et OS X, Noyau Linux, Android, Arduino et RFduino et OMNET++).
- Une mise en œuvre partielle du protocole de gestion interopérable.
- Un serveur HTTP simple pour afficher la configuration interne du relais.
- Quelques extensions intéressantes.

4.2. Description conceptuelle de l'intégration CCN-lite/OMNeT++

OMNeT ++ représente une approche Framework. Au lieu de fournir directement des composants de simulation pour les réseaux informatiques ou d'autres domaines, il fournit les machines de base et les outils pour écrire de telles simulations. Les domaines d'application spécifiques sont pris en charge par différents modèles de simulation et frameworks tels que Mobility Framework ou INET Framework. Ces modèles sont développés indépendamment de OMNeT ++.

INET Framework est une bibliothèque des modèles open sources pour l'environnement de simulation OMNeT ++. Il fournit des protocoles, des agents et d'autres modèles pour les chercheurs et les étudiants qui travaillent avec les réseaux de communication. INET est particulièrement utile lors de la conception et la validation de nouveaux protocoles, où en explorant des nouveaux ou exotiques scénarios [14].

La figure 23 représente l'intégration de CCN-lite avec OMNeT++ et INET Framework qui est nécessaire pour faire la simulation. Parmi les dépendances évidentes : OMNeT++ (> v4.2.2) simulateur et INET Framework (> v1.99.4.).

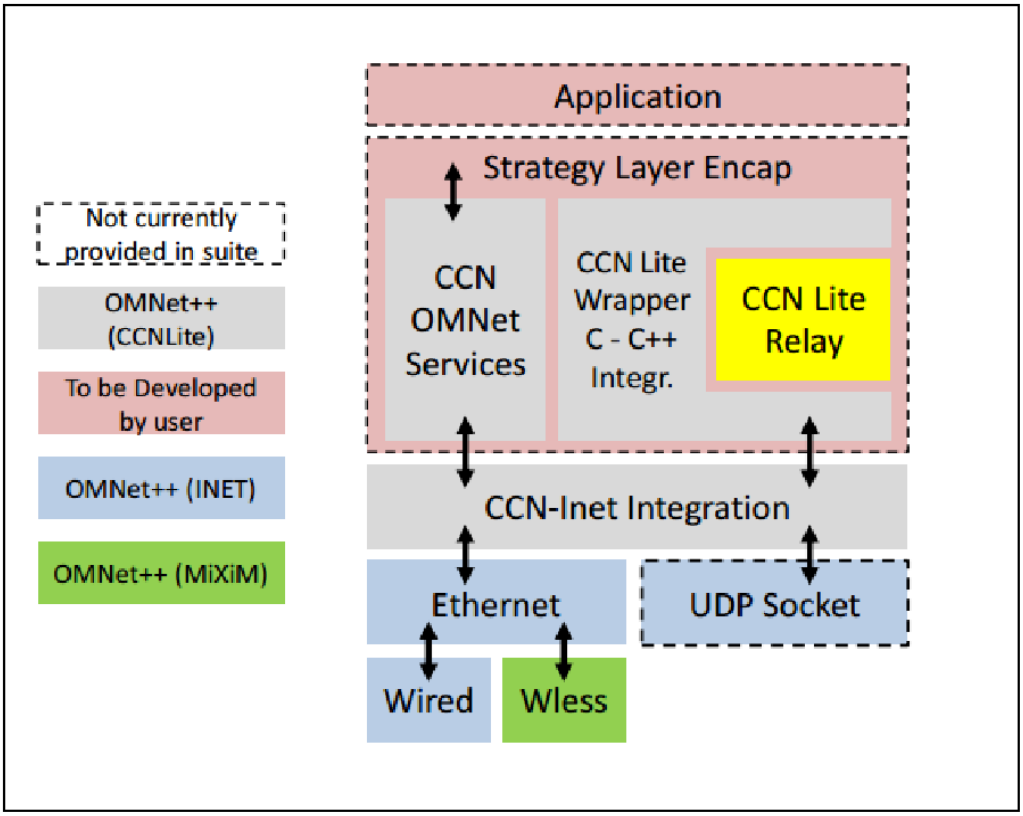

Figure 23: Schéma d'intégration CCN-lite avec OMNeT++. [11]

4.3. Diagramme de classe des composants CCN-lite/OMNeT++

Les classes les plus importantes dans le package CCN-lite sont celles du répertoire **ccnlite/src/** (voir la Figure 24 et Figure 25) :

• **ccn-lite / :** Dossier contient l'implémentation réelle de CCN-lite relais en C.

• **CcnCore. {Cc, h} :** Intégration CCN-lite (C-C ++).

• **Ccn. {Cc, h, Ned} :** Services OMNeT++.

• **CcnInet. {Cc, h, Ned} :** Intégration OMNeT++ / INET Framework.

• **CcnAdmin. {Cc, h, Ned} :** Administrateur de Scénario.

• **Parser. {Cc, h} :** Analyseur de scénario.

• **CcnPacket_m. {Cc, h} :** Conteneur extensible pour les paquets CCN échangés via des nœuds CCN dans OMNeT++.

• **CcnAppMessage_m. {Cc / h / msg} :** C'est un message échangé entre la couche CCN et la couche ci-dessus (stratégie ou application).

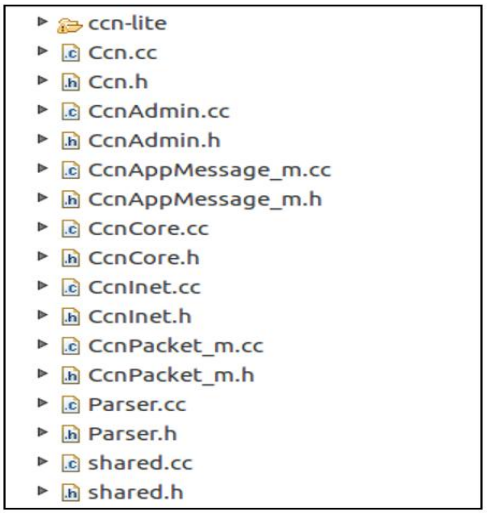

Figure 24: Les fichiers de CCN-lite

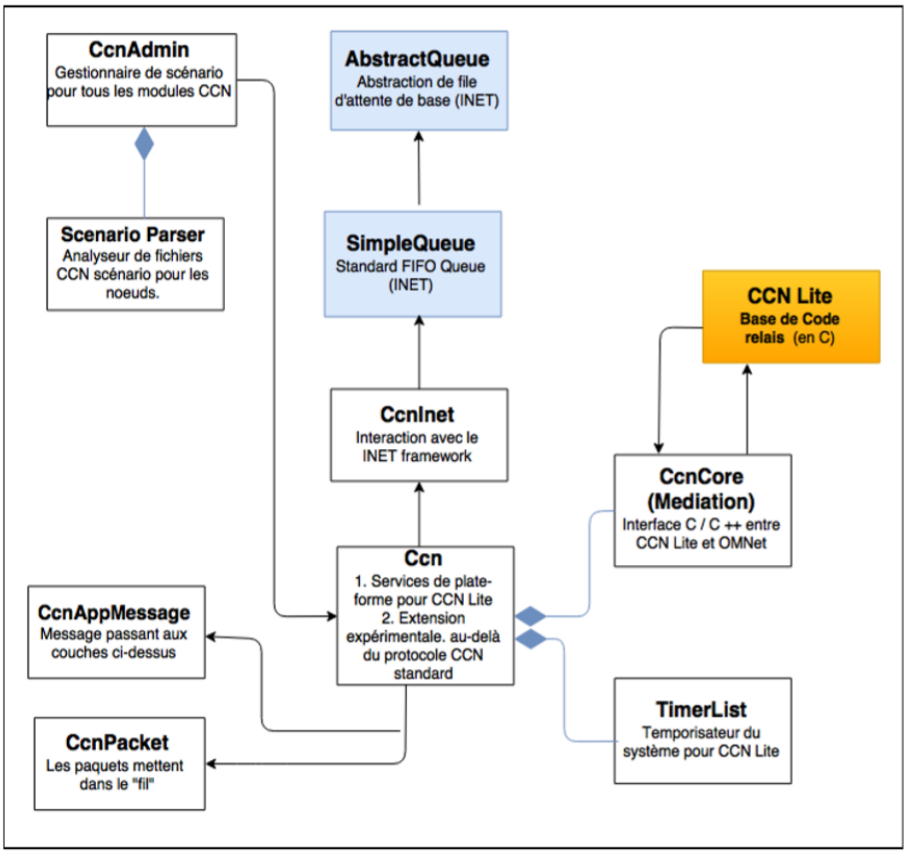

Figure 25: Diagramme de classe UML des Composants CCN-lite/OMNeT++ [11].

5. Environnement de travail

Nous allons détailler les environnements utilisés dans la réalisation de notre simulation.

5.1. Environnement matériel

La simulation a été réalisée sur un ordinateur Lenovo dont la configuration est :

Processeur	Intel Core i5 CPU @ 2.5 GHz
Mémoire	4GB DDR4
Disque dur	1 TO

5.2. Environnement logiciel

Après plusieurs essaies dans différents systèmes d'exploitation et plusieurs version d'OMNeT++ et INET, notre simulation a été réalisée dans l'environnement logiciel suivant:

- Système d'exploitation : Linux Distribution Ubuntu 16.04.

- Le simulateur OMNeT++ 4.5.
- La plateforme INET v 3.0.2.

6. Simulation

Avant de présenter notre scénario d'attaque, nous commençons par expliquer comment créer un CCN en utilisant CCN-lite et OMNet++.

6.1. Création d'un CCN

Un CCN est caractérisé par le déploiement de plusieurs nœuds. Pour notre simulation, il fallait commencer par :

✓ **Etape1 :** créer un CCN comme suit (figure 26):

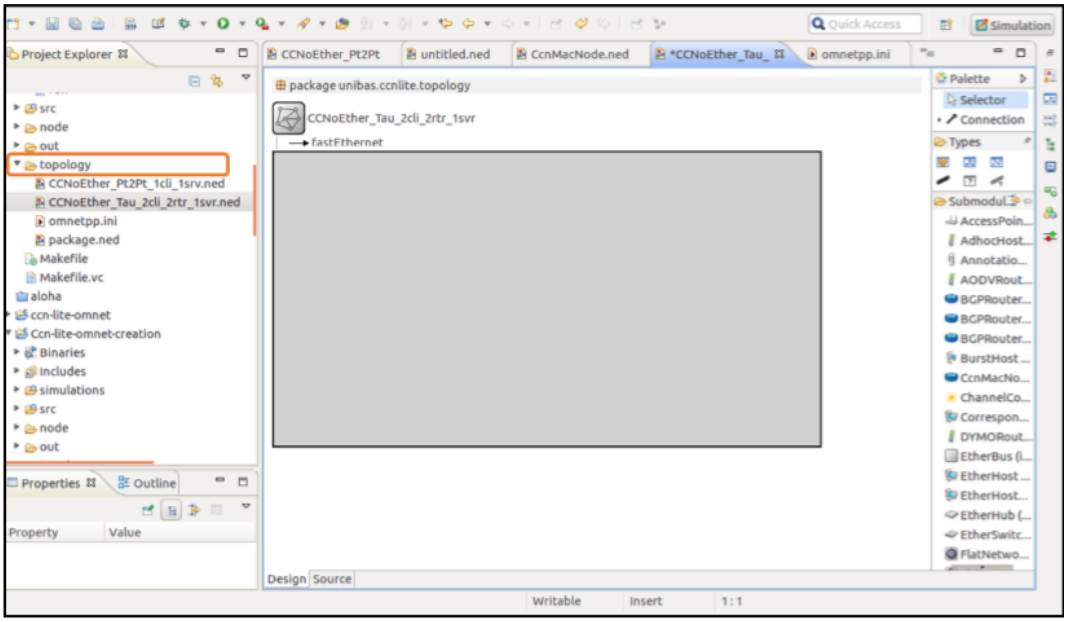

Figure 26: Création de Topologie de réseaux CCN.

✓ **Etape 2 :** Créer les nœuds nécessaires pour notre topologie CCN : **server1, server2, client1, client2** et **attacker** (figure 27).

Le fichier graphique ".ned" (figure 28 et figure 29) illustre les différents modules utilisés par un nœud CCN.

Tous les nœuds ont la même structure « CcnMacNode » (figure 30).

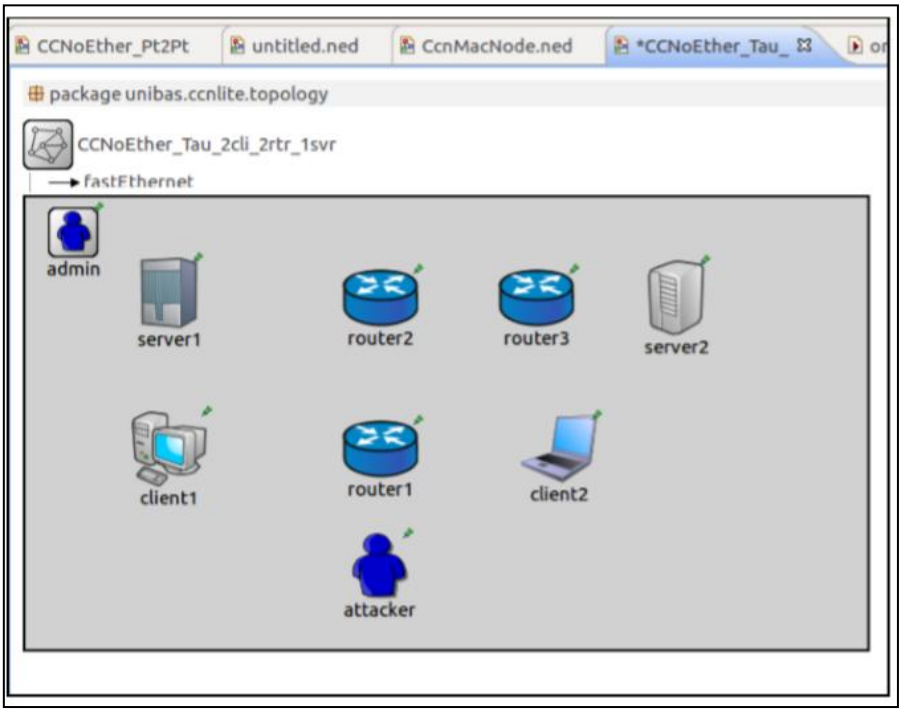

Figure 27: Création de Topologie de réseaux CCN.

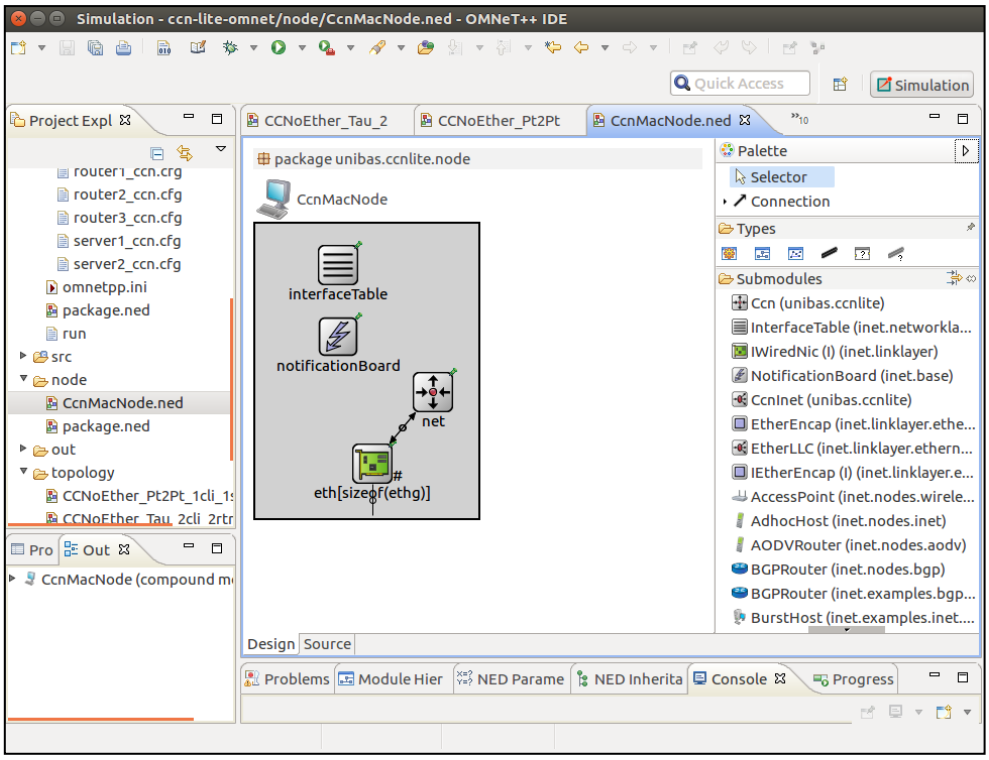

Figure 28: Interface graphique de simulateur OMNeT++ avec CCN-lite.

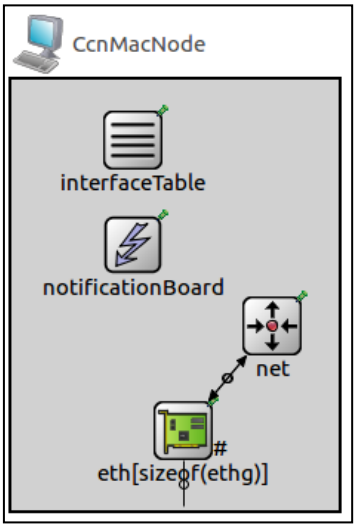

Figure 29: Fichier graphique ".ned" du nœud CCN.

CCNoEther_Tau_2cli_2rtr_1sv | *CcnMacNode.ned | server2_ccn.cfg | serv

```
package unibas.ccnlite.node;
import unibas.ccnlite.Ccn;
import inet.networklayer.common.InterfaceTable;
import inet.linklayer.IWiredNic;
//import inet.mobility.IMobility;
import inet.base.NotificationBoard;
//import inet.mobility.models.StationaryMobility;
module CcnMacNode
{
    parameters:
        @display("i=device/pc");
        @node;
        @labels(node,ethernet-node);

    gates:
        inout ethg[] @labels(EtherFrame-conn);

    submodules:
        // events pub-sub in a cross-layer fashion
        notificationBoard: NotificationBoard {
            parameters:
                @display("p=82,112");
        } // --- Network Layer
        net: Ccn {
            @display("p=176,167");
        }

        interfaceTable: InterfaceTable {
            parameters:
                @display("p=82,41");
        }// --- Link layer Wired Ethernet NICs
        eth[sizeof(ethg)]: <default("EthernetInterface")> like IWiredNic {
            parameters:
                @display("p=116,238,row,90;q=txQueue");
        }

    connections allowunconnected: // --- connect NICs to outside world and the
            for i=0..sizeof(ethg)-1 {
            ethg[i] <--> eth[i].phys;
            eth[i].upperLayerOut --> net.ifIn++;    // NOTE: in older versions
            eth[i].upperLayerIn <-- net.ifOut++;
```

Design | Source

Figure 30: Fichier de code source ".ned" d'un nœud CCN.

✓ **Etape 3 :** Gérer les connexions entre les nœuds au niveau de la structure dans **"Connections"** du fichier de Topologie **".ned"**. Un réseau CCN est composé du module nœud décrit précédemment ainsi d'un canal pour la communication (fastEthernet) entre nœuds (figure 31).

```
connections:

    client1.ethg[0] <--> fastEthernet <--> router2.ethg[0];
    router2.ethg[3] <--> fastEthernet <--> router1.ethg[0];
    router1.ethg[1] <--> fastEthernet <--> server1.ethg[0];
    router1.ethg[2] <--> fastEthernet <--> hacker.ethg[0];
    client2.ethg[0] <--> fastEthernet <--> router2.ethg[1];
    server2.ethg[0] <--> fastEthernet <--> router3.ethg[0];
    router3.ethg[1] <--> fastEthernet <--> router2.ethg[2];
    hacker.ethg[1] <--> fastEthernet <--> router2.ethg[4];
    hacker.ethg[2] <--> fastEthernet <--> router3.ethg[2];
```

Figure 31: Gestion des Connexion pour tous les nœuds CCN.

✓ **Etape4 :** Créer les fichiers de rôle pour chaque nœud dans le scénario qui sont fournis dans le fichier ".cfg " inclus. Ces fichier contient (figure 32) :

- **[eInterestMode]** : La table PIT qui contient **ContentName** (le nom de contenu), **StartChunk** (debut de segement), **ChunksCount** (nombre des segments) et **RequestTime** (temp de réponse).
- **[ePreCacheMode]** : Le CS (le cache) qui contient **ContentName** (le nom de contenu), **StartChunk** (debut de segement), ChunksCount (nombre des segment) et **UpdateTime** (temps de réponse).
- **[eFwdRulesMode]** : La table FIB qui contient **ContentPrefix** (préfix du nom de contenu), **NextHop** (interface de nœud suivant), **AccessFrom** (interface de sortie) et **UpdateTime** (temps de mise a jour).

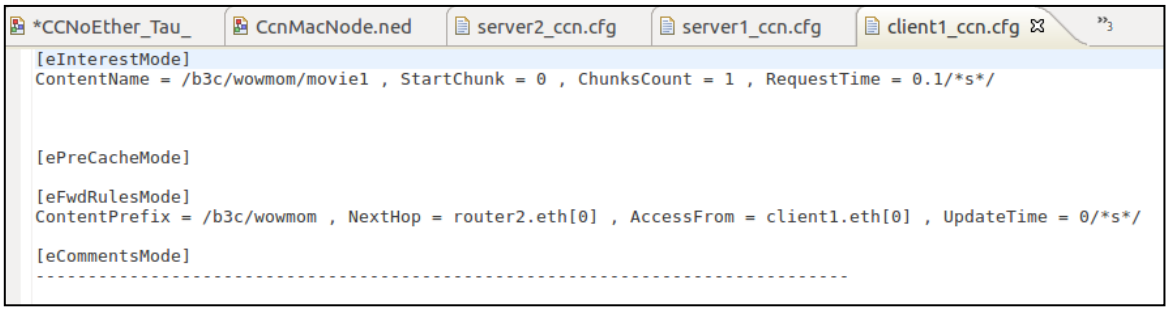

Figure 32: Fichier scénario de client1.

- ✓ ***Etape5 :*** Créer le fichier "**.ini**" (qui est lié au fichier NED) (figure 33) qui permet à l'utilisateur d'initialiser les paramètres des différents modules ainsi la topologie du réseau.

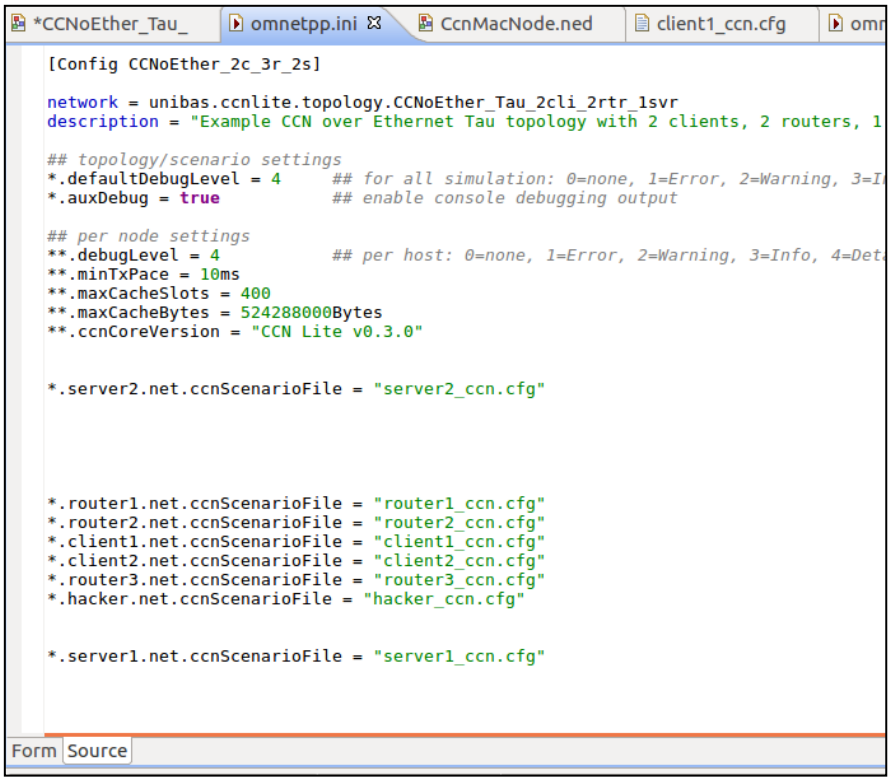

Figure 33: Fichier omnetpp.ini.

6.2. Scénario d'attaque

6.2.1. Description du scénario

La Figure 34 représente Un scénario d'attaque dans un réseau CCN simple composé de trois routeurs, un attaquant, deux clients normaux et deux serveurs. La création est faite comme décrit dans la section précédente.

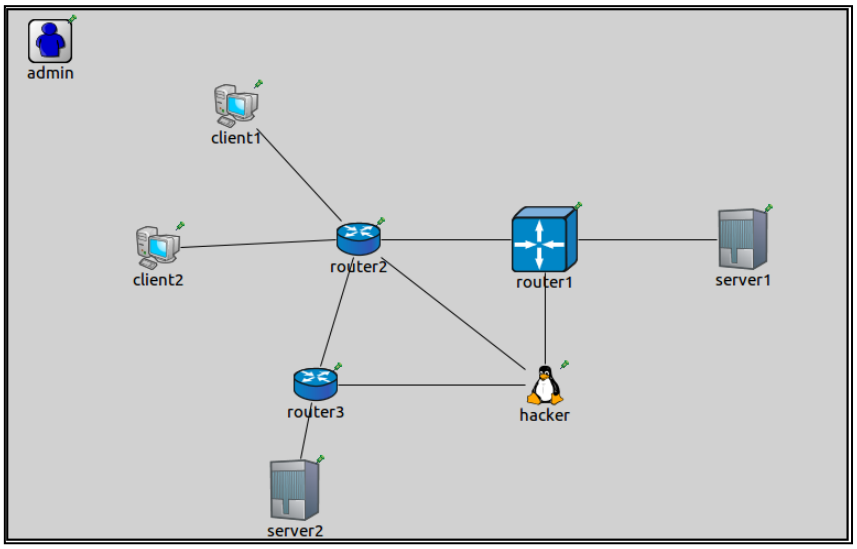

Figure 34: La topologie du scénario

Nous supposons qu'un administrateur malicieux qui va orienter les routeurs vers une fausse destination qui est celle d'un attaquant lors des mises à jours en accédant aux fichiers routeur_ccn.cfg et en modifiant le fichier **eFwdRulesMode** (voir la figure 35).

```
[eInterestMode]

[ePreCacheMode]

[eFwdRulesMode]
ContentPrefix = /b3c/wowmom , NextHop = hacker.eth[1] , AccessFrom = router2.eth[4] , UpdateTime = 0/*s*/
ContentPrefix = /b3c/wowmom , NextHop = router3.eth[1] , AccessFrom = router2.eth[2] , UpdateTime = 0/*s*/

[eCommentsMode]
--------------------------------------------------------------------------------
comments go here
```

Figure 35: Le fichier eFwdRulesMode après la mise à jour.

Le rôle de l'attaquant est celui d'espionner et de censurer les paquets circulants entre le consommateur et le producteur en passant par lui avant d'aller vers le prochain le routeur prévu (voir la figure 36).

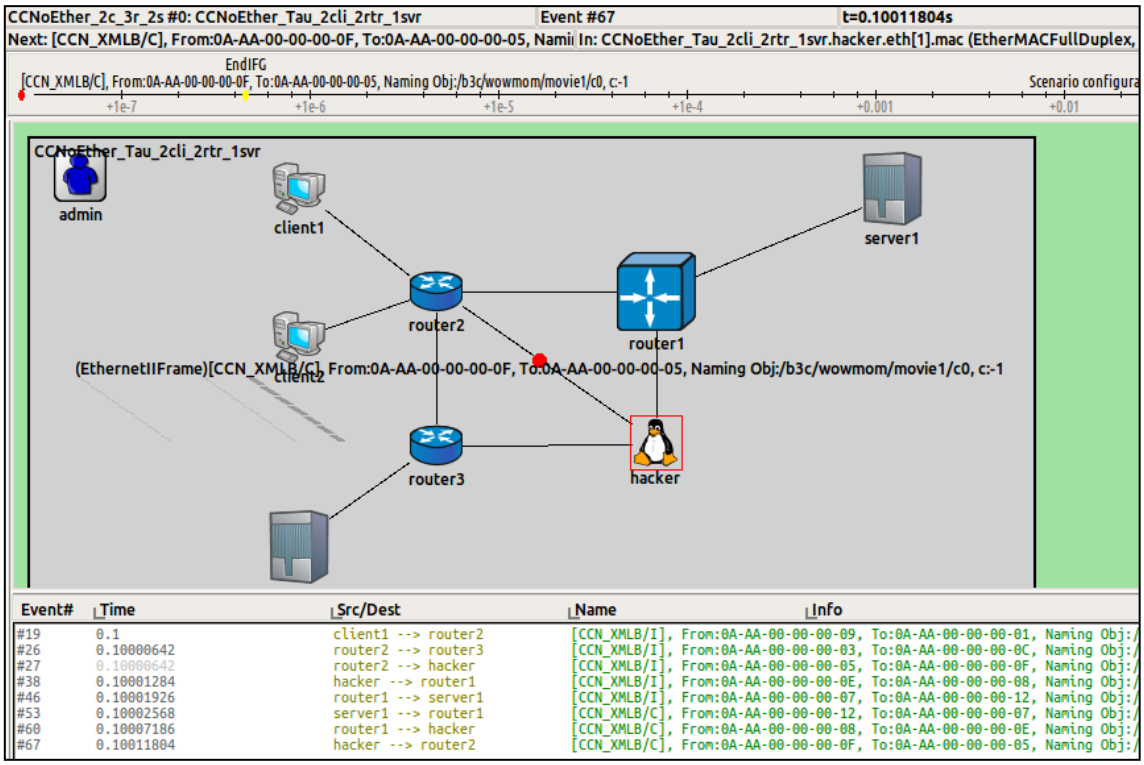

Figure 36: Les paquets passant par l'intercepteur.

7. Implémentation de l'algorithme d'anonymisation

L'algorithme d'anonymisation est lancé après avoir crée notre topologie, il permet de (dans l'ordre) :

1. Signer les paquets de contenu avec l'infrastructure PKI par le serveur.
2. Vérification de la signature par le consommateur.
3. Crypter les contenus avec le chiffrement asymétrique RSA au niveau du serveur.
4. Décryptage du contenu par le consommateur.

Cet algorithme est implémenté dans les classes suivantes :

- **CcnCore.h** : contient la déclaration des structures de données et les fonctions nécessaires (voir Figure 37).

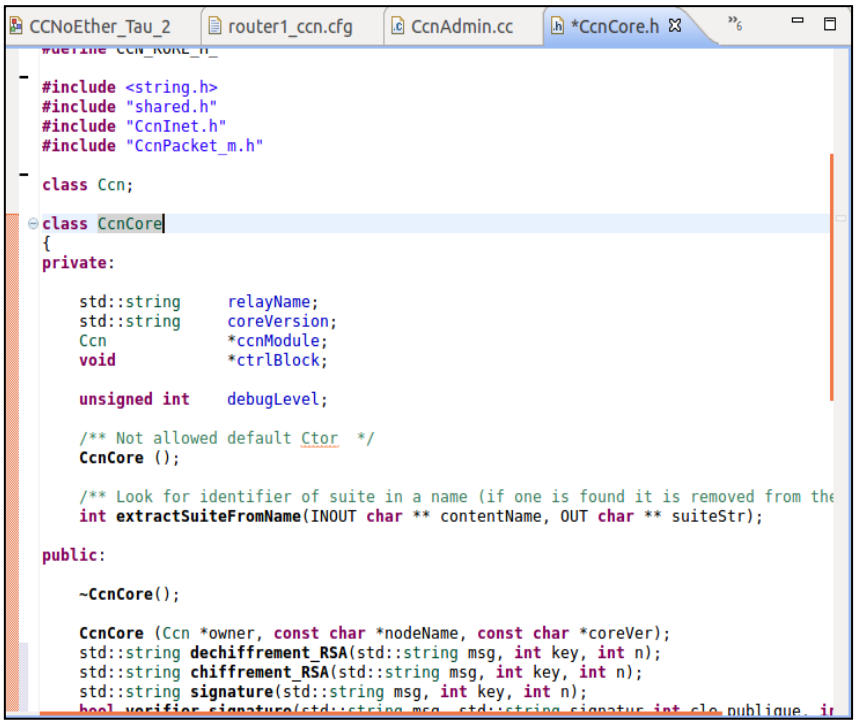

Figure 37: Le prototype des fonctions définies dans l'algorithme d'anonymisation

- **CcnCore.cc** contient le code source des fonctions **CcnCore.h** (Figure 38).

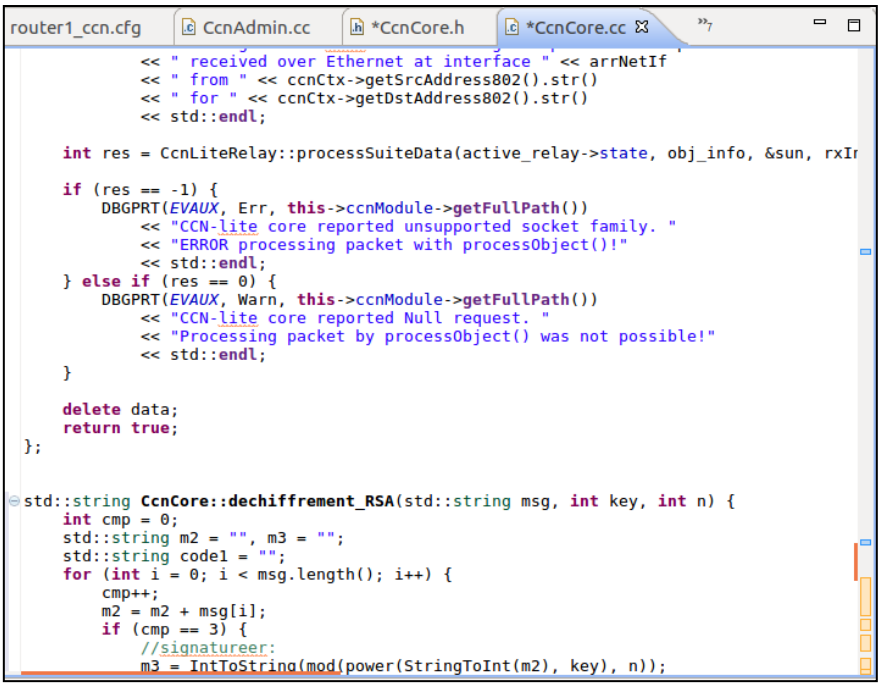

Figure 38: aperçu des fonctions d'Anonymisation.

8. Conclusion

Pour la confidentialité et la sécurité de la vie privée plusieurs défis ont été menés pour la lutte contre la violation de ces derniers, parmi ces attaques les plus connues nous citons l'attaque d'interception qui a fait l'objet de notre étude. Pour atténuer le risque de cette attaque et renforcer le système de sécurité dans les NDNs/CCNs nous avons proposé une approche que nous avons appelé l'approche de confidentialité asymétrique qui a pour but d'assurer la confidentialité et l'intégrité des contenus à l'aide d'une encapsulation chiffrée et signée des contenus.

Pour la simulation des réseaux plusieurs techniques de modélisation ont été créées, qui ont pour but de voir le comportement des nœuds et leur fonctionnement. Nous avons utilisé dans notre travail le simulateur « OMNET++ ». Nous avons présenté en premier lieu l'environnement de ce dernier ensuite nous avons fait une description pour le package CCN-lite. Ce dernier nous a permis de faire des simulations d'attaques d'interception dans les NDNs et d'implémenter l'algorithme de confidentialité pour faire face à une telle attaque.

CONCLUSION GENERALE

Internet a connu un immense succès mondial, dans son ensemble, malgré ça il montre clairement des signes de vieillesse depuis le changement du monde de communication qui exige des services à forte intensité d'information, des exaoctets de contenu créés et consommés quotidiennement sur le Web ainsi qu'une ménagerie de périphériques mobiles qui y sont connectés. Pour suivre le rythme de ces changements et faire évoluer l'Internet vers l'avenir, un certain nombre d'efforts de recherche visant à concevoir de nouvelles architectures Internet ont pris leur essor au cours des dernières années. Le réseau de données nommées ICN est l'un de ces efforts qui illustre l'approche centrée sur le contenu pour la mise en réseau. Il se concentre sur le contenu afin de fournir une distribution de contenu évolutive et efficace ce qui le rend unique. Il dépend principalement de la dénomination indépendante de la localisation, de la mise en cache dans le réseau et du routage basé sur les noms.

Il existe de nombreuses propositions pour les architectures ICN comme DONA, NetInf, NDN et PURSUIT. Après la présentation de ces architectures, nous nous sommes orientées vers l'architecture du NDN (Named Data Networking) où nous avons étudiés son système de routage et de sécurité. L'amélioration des aspects de sécurité et de confidentialité dans NDN mène à de nouveaux défis liés au routage centré contenu tels que les attaques d'infrastructure, de source, de Blocage mobile, d'innodation (Flooding), de minutage (timing), de brouillage (Jamming), de détournement (Hijacking) et d'interception. Parmi ces attaques, nous nous sommes intéressées par l'attaque d'interception. Dans un scénario spécifique et concret d'attaque d'interception dans NDN, nous avons démontré la possibilité d'attaque d'interception où les attaquants font usage de la règle d'acheminement des paquets d'intérêt de NDN par la table FIB et se mettent en intermédiaire. Par la suite l'attaquant verra tout trafic circulant entre le consommateur et le producteur de contenu.

Pour lutter contre les attaques d'interceptions, plusieurs solutions ont été proposées, mais qui

ont un coût élevé. D'après notre étude, nous avons conçu une solution que nous avons appelés protocole de confidentialité asymétrique qui permet, entre autres, de renforcer la sécurité, de garantir la confidentialité et l'intégrité à l'aide des signatures et d'une encapsulation chiffrée des paquets de contenu. Afin de valider notre approche, nous avons intégré CCN-lite avec OMNeT++ (> v4.2.2) et INET Framework (> v1.99.4.). CCN-lite est une implémentation réduite et légère des protocoles CCN notamment le routage centré contenu. Elle nous a permis de créer un CCN, de simuler le scénario d'attaque d'interception et de tester notre algorithme.

Au moment de la rédaction de ce mémoire, nous rencontrons toujours des problèmes liés à l'implémentation de l'algorithme dans CCN-lite. Donc, nous n'avons pas pu voir son exécution mais nous avons donné une vue sur son fonctionnement prévu. Nous pensons après plusieurs essaies qu'il est trop difficile d'implémenter l'algorithme dans CCN-lite (code source qui ne l'appartiennent pas), mais ce n'est pas impossible dans le futur proche. Une fois implémenté, plusieurs tests devront être faits sur l'algorithme pour ajuster ses paramètres et calculer ses performances. Ensuite, nous suggérons, comme perspectives, de:

- Intégrer d'autres solutions de lutte contre l'interception et de comparer avec la solution proposée.
- Simuler d'autres types d'attaques liés au routage.

BIBLIOGRAPHIE

[1] E. G. Z. M. &. H. H. S. AbdAllah, «A Security Framework for ICN Traffic Management,» 2018.

[2] K. &. J. S.-H. Hasan, «A Cluster-Based Content Management Framework for Information-Centric Networking,» n° %1100.1109, 2018.

[3] W. You, «A Content-Centric Networking Node for a Realistic Efficient Implementation and Deployment,» Bretagne, 2014.

[4] M. S. K. K. A. B. R. R. N. &. Q. A. Akbar, «Information-Centric Networks: Categorizations, challenges, and classifications,» USA, 2014.

[5] E. G. H. H. S. &. Z. M. AbdAllah, «Enquête sur les attaques de sécurité dans les réseaux centrés sur l'information,» 2015.

[6] T. M. S. M. Reza Tourani, «Sécurité, confidentialité et contrôle d'accès dans les réseaux centrés sur l'information: une enquête,» Mexique, 2017.

[7] 25 Février 2006. [En ligne]. Available: http://www.omnetpp.org/doc/INET/neddoc/, .

[8] S. O. A. A. A. B. Z. L. Z. a. L. W. N. A. Hoque, «A Secure Link State Routing Protocol for NDN,» 2016.

[9] N. Benkirane, «La gestion du trafic dans les réseaux orientés contenus,» Paris, 2014.

[10] «CCN-lite Project and Community,» [En ligne]. Available: http://ccn-lite.net/.

[11] «CCN-lite en GitHub,» [En ligne]. Available: https://github.com/cn-uofbasel/ccn lite/blob/master/doc/internal/omnetpp-getting-started.pdf.

[12] F. R. T. C. François-Xavier Aguessy, «Security analysis of the virtualized NDN architecture,» Janvier 2016.

[13] F. R. T. C. E. M. d. O. W. M. G. D. T. N. R. C. T. C. X. M. François-Xavier Aguessy, «Security analysis of the virtualized NDN architecture,» Paris, 2016.

[14] «INET Framework,» [En ligne]. Available: https://inet.omnetpp.org/Introduction.html.

[15] P. G. G. T. E. U. Steven Diabenedeto, «AnonymousNamedDataNetworkingApplication».

[16] N. H. B. R. L. B. Van Jacobson Diana K. Smetters James D. Thornton Michael F. Plass, «Networking Named Content,» USA.

[17] 2 MARS 2007. [En ligne]. Available: http://www.omnetpp.org/,.

[18] R. M. Walid Miloud, «Etude et manipulation des mécanismes de sécurité pour le routage centré contenu,» 2018.

[19] D. Pointcheval, «Le chiffrement Asymétrique et la Sécurité Prouvée,» Paris, 2002.

[20] A. Bendouma, «Développement d'une infrastrecture de gestion de clés de cryptage dans les réseaux AD HOC véhiculaires,» Québec, 2017.