RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

Ministère de L'enseignement Supérieur et de la Recherche Scientifique

Université MOULOUD MAAMERI Tizi_Ouzou

Faculté de génie électrique et informatique

Mémoire de projet de fin d'étude en vue de l'obtention du Diplôme de Master

Domaine : électronique

Spécialité : réseaux et télécommunications

THÈME

Sécurisation des plates-formes WEB à l'aide de certificats.

Présenté par:

Mr ABDELLAOUI Omar **Mme BOUSSADI Amel Djouher**

Promoteur : ALOUACHE Djamal

Co-promoteur : AMALOU ABDENOUR

Promotion 2017-2018

Remerciements

Avant toute chose, nous remercions Dieu, le tout puissant, pour nous avoir donnés la force, la volonté, et la patience durant toutes nos années d'étude.

Nous tenons avant tout, à exprimer notre immense gratitude envers notre promoteur monsieur le Professeur ***ALLOUACHE Djamal,*** *pour sa rigueur, son esprit scientifique et perfectionniste et la confiance qu'il nous a accordée. Qu'il trouve ici le témoignage de notre reconnaissance.*

Nous remercions également notre co-promotreur Monsieur ***AMALOU ABDNOUR*** *pour sa disponibilité, sa rigueur et ses compétences scientifiques et pour le thème passionnant qu'il nous a confié. Qu'il trouve ici l'expression de notre vive reconnaissance et de notre profond respect.*

Nous remercions aussi monsieur le président LAZRI Mourad d'avoir accepté présider notre jury.

Nos remerciements vont également à monsieur l'examinateur OUALOUCHE Fethi d'avoir accepté d'examiner ce travail.

Nos vifs remerciements vont aussi à toutes les personnes qui ont contribué de prêt ou de loin au bon déroulement de ce travail.

À cœur vaillant rien d'impossible

À conscience tranquille tout est accessible

Quand il y a la soif d'apprendre
Tout vient à point à qui sait attendre

Quand il y a le souci de réaliser un dessein
Tout devient facile pour arriver à nos fins

Malgré les obstacles qui s'opposent
En dépit des difficultés qui s'interposent

Les études sont avant tout
Notre unique et seul atout

Ils représentent la lumière de notre existence
L'étoile brillante de notre réjouissance

Comme un vol de gerfauts hors du charnier natal
Nous partons ivres d'un rêve héroïque et brutal

Espérant des lendemains épiques
Un avenir glorieux et magique

Souhaitant que le fruit de nos efforts fournis
Jour et nuit, nous mènera vers le bonheur fleuri

Aujourd'hui, ici rassemblés auprès des jurys,
Nous prions dieu que cette soutenance
Fera signe de persévérance
Et que nous serions enchantés
Par notre travail honoré

MERCI

Sommaire

Chapitre II : Sécurité eu cryptographie

Chapitre III : Services de certificat AD CS

Chapitre IV : Réalisation pratique

Liste des abréviations

AD CS: Active Directory Certificate Services.

AD DS: Active Directory Domain Services.

AD FS: Active Directory Federation Service.

AD RMS: Active Directory Rights Management Services.

ACL: Access Control List

AES: Advanced Encryption Standard.

AIA: Authority Information Access.

API Microsoft Crypto:

CA: Certificate Authority.

CDP: CRL Distribution Points.

CFB: Cipher Feedback Block

CRT: Chinese Remainder Theorem.

CPU: Central Processing Unit.

CRL: Certificate Revocation Lists.

DES: Data Encryption Sandard.

DNS: Domain Name System.

DHCP: Dynamic Host Configuration Protocol.

ECB: Electronic Code Book

EFF: Electronic Frontier Foundation.

EFS: Encrypting File System.

GSM: Global System for Mobile

HTTPS: Hyper Text Transfer Protocol.

IDEA: International Data Encryption Algorithm

IIS: Internet Information Services.

LAN: Local Area Network.

NIST: National Institute of Standards and Technology.

NTFS: New Technology File System

OS: Operating System

PKI: Public Key Infrastructure.

RSA: Revenu de solidarité active.

SQL: Structured Query Language.

SSL: Secure Socket Layer

TDES ou 3DES : Triple Data Encryption Sandard.

URL: Uniform Resource Locater.

VPN IPsec: Virtual Private Network Internet Protocol Security.

VSS: Volume Shadow Copy Service.
WAN: Wide Area Network.

WEP: Wired Equivalent Privacy

WID: Windows Internal Database.

XOR: eXclusive OR

Liste des figures

Liste des tableaux

Introduction générale

Les réseaux informatiques occupent une place de plus en plus importante et deviennent un outil incontournable dans les sociétés modernes. Ils procurent d'énormes avantages aux entreprises et facilitent les différentes transactions et opérations en un temps record. En effet, depuis l'avènement relativement récent du règne informatique, les systèmes matériels et logiciels ont reçu un intérêt particulier et une utilisation accrue dans tous les domaines de la vie allant de la vie courante aux domaines les plus techniques.

L'internet constitue un outil incontournable pour permettre aux entreprises d'accroître leurs performances que ce soit en matière de productivité qu'en matière de rentabilité. Toutefois, ce phénomène n'empêche pas de considérer le fait qu'à force d'être connectées en permanence à Internet, bon nombre d'entreprises s'exposent en même temps aux différentes formes d'attaques cybercriminelles, plus particulièrement au piratage informatique si elles ne prennent pas suffisamment le soin de protéger leurs données en renforçant davantage la sécurisation de leur système d'information.

En effet, on assiste actuellement, à une explosion de la criminalité informatique, notamment des intrusions dans les réseaux internes des sociétés par des personnes étrangères, usurpation d'identité, vole d'informations confidentielles pour effectuer des transactions bancaires ou autres opérations financières et surtout l'accès à distance, aux données de l'entreprise par ces propres employés (VPN – avec une clé de certificats) ou par des ex-employés qui en connaissent les détails ce qui présente le danger le plus redoutable pour les responsables d'entreprises.

Certaines attaques informatiques exploitent des failles trouvées dans les différents systèmes qui composent le réseau des entreprises. Les commanditaires de ces attaques informatiques (les hackers) chiffrent certaines données des systèmes des entreprises et prennent alors le contrôle. Ce qui paralyse les entreprises ciblées et leur coûte lourdement tant en argent qu'en temps. Le préjudice financier peut atteindre plusieurs dizaines de millions d'euros par an. Ces attaques peuvent même mener à la faillite et menacer la survie de ces entreprises. Les hackers demandent alors des rançons pour décoder les systèmes qu'ils avaient auparavant chiffrés.

Les piratages informatiques (cyberattaques) se multiplient. Chaque année, une entreprise sur quatre environ serait victime de logiciels de rançons informatiques (ransomware). Les proies les plus vulnérables sont les entreprises qui sont souvent mal protégées au niveau des systèmes

d'exploitation et des antivirus. Les attaques sont d'autant plus difficiles à prévenir qu'elles arrivent parfois par un simple mail.

Face au piratage informatique, les entreprises doivent, en effet, élaborer et mettre en place un système de protection parfaitement infaillible pour renforcer la sécurité des entreprises. A mesure que l'industrie se numérise, la sécurité des données devient un enjeu prioritaire. De nombreuses solutions de protection, aussi différentes les unes des autres, ont vu le jour. Ces dernières, afin qu'elles soient efficaces, doivent être bien choisies, bien placées et bien configurées suivant les caractéristiques du réseau mis en place.

Dans ce contexte et au cours de ce mémoire, nous nous proposons de concevoir et de simuler une plateforme WEB d'une société à l'aide d'un système IIS. Nous nous proposons par la suite, de la sécuriser à l'aide d'un certificat SSL afin d'éviter toute infraction ou violation de notre système informatique. Une partie pratique est également présentée afin de tester la compétence et l'efficacité de notre système de sécurisation.

Le présent mémoire s'articule autour de quatre (04) chapitres :

- Le premier chapitre est consacré à la présentation de quelques notions théoriques indispensables sur l'architecture Client/ serveur.
- Le deuxième chapitre traite de la sécurité et de la cryptographie des réseaux informatiques pour une meilleure sécurité de l'information et une plus grande confidentialité des données.
- Le troisième chapitre présente le concept services de certificat AD CS qui permet de vérifier l'identité de chaque entité, composant du réseau, individu, système, ou autres à l'aide d'un certificat issu d'une autorité de certification approuvée.
- Le quatrième chapitre fera l'objet des différents outils utilisés pour le développement de notre application et une illustration de certaines de ses fonctionnalités. Enfin, nous terminerons notre mémoire par une conclusion générale.

I.I. Préambule

Un système réparti permet à des utilisateurs, situés à des endroits différents, de coopérer et de mettre en commun leurs ressources. L'accès à distance aux ressources requiert un modèle d'interaction entre les utilisateurs et les ressources répartis. Le partage des ressources est intéressant parce qu'il permet, entre autres, de réduire les coûts, d'échanger de l'information, de diffuser rapidement des données et de collaborer à distance. Un gestionnaire de ressources est un logiciel responsable de l'administration d'un type de ressources. Il possède une interface de télécommunications à travers laquelle s'effectuent l'accès et la mise à jour des ressources par les utilisateurs. Le gestionnaire met également en œuvre les politiques d'accès propres à chaque type de ressources (ex: les imprimantes). Les trois concepts de base des systèmes répartis (ressource, gestionnaire de ressources et utilisateur de ressources) peuvent être structurés et mis en relation suivant deux grands modèles, le modèle client- serveur et le modèle des composants répartis.

I.2. Concepts des réseaux locaux

Un réseau, nous l'avions compris, permet de connecter des ordinateurs entre eux. Mais les besoins sont très divers, depuis le réseau domestique ou d'une petite entreprise jusqu'aux réseaux des grandes sociétés.

Il existe deux approches fondamentalement différentes, dont l'une peut facilement évoluer vers d'autres.

I.2.1. Poste à poste « Peer to Peer »

Les postes de travail sont simplement reliés entre eux par le réseau. Aucune machine ne joue un rôle particulier, chaque poste peut partager ses ressources avec les autres postes et chaque utilisateur définira l'accès à ses ressources. Dans ce cas, il n'y'a pas obligatoirement d'administrateur attitré.

Tableau 1 : Avantages et inconvénients du Peer to Peer.

Avantages	Inconvénients
• Il est facile de mettre en réseau des postes qui étaient au départ isolés • Chaque utilisateur peut décider de partager l'une de ses ressources avec d'autres postes • Dans un groupe de travail, l'imprimante peut être utilisée par tous • Cette méthode est pratique et peu couteuse pour créer un réseau domestique	• Chaque utilisateur a la responsabilité du fonctionnement du réseau • Les outils de sécurité sont très limités • Si un poste est éteint ou s'il se « plante », ses ressources ne sont plus accessibles • Le système devient ingérable lorsque le nombre de poste augmente • Lorsqu'une ressource est utilisée sur une machine, l'utilisateur de cette machine peut voir ses performances diminuer

I.2.2. Définition du Client/serveur [1]

Le modèle client-serveur s'articule autour d'un réseau auquel sont connectés deux types d'ordinateurs le serveur et le client. Le client et le serveur communiquent via des protocoles. Les applications et les données sont réparties entre le client et le serveur de manière à réduire les coûts. Le client-serveur représente un dialogue entre deux processus informatiques par l'intermédiaire d'un échange de messages. Le processus client sous-traite au processus serveur des services à réaliser. Les processus sont généralement exécutés sur des machines, des OS et des réseaux hétérogènes.

I.2.2.1. Présentation de l'architecture client/serveur [1]

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client FTP, client de messagerie, etc… lorsque l'on désigne un programme, tournant sur une machine cliente, capable de traiter des informations

qu'il récupère auprès du serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client messagerie il s'agit de courrier électronique).

I.2.2.2. Paradigme Client/serveur

La plupart des applications réseau modernes se fondent sur le paradigme Client/serveur. Le Client/serveur est un mode de dialogue entre deux entités. La première appelée client demande des services à l'entité serveur qui les exécute et envoie les résultats en retour. Le vocable Client/serveur est souvent employé dans deux contextes; physique et logique. Dans le premier contexte; clients et serveurs représentent des systèmes physiques (machines) tandis que dans le second, les qualificatifs clients et serveur relèvent du domaine logiciel et donnés donc à des processus, dans ce cas, on peut avoir une application Client/serveur qui tourne sur une seule machine physique qui assurera le rôle de client et serveur en même temps.

I.2.2.3. Caractéristiques du Client/Serveur [2]

Bien que le Client/ Serveur soit une forme d'informatique distribuée, il n'est supposé d'appeler systèmes Client/ Serveur que ceux qui partagent les caractéristiques suivantes :

- ✓ **Service :** Le modèle Client/ Serveur est essentiellement une relation entre des processus tournant sur des machines séparées. Le processus Serveur est un fournisseur de services. Le Client est le consommateur de ces services. Le modèle Client/serveur établit ainsi une séparation claire des rôles à partir de la notion du service.
- ✓ **Partage des ressources :** Un serveur est sensé pouvoir traiter plusieurs clients à la fois et contrôler leurs accès aux ressources.
- ✓ **Asymétrie des protocoles :** La relation entre clients et serveur est de type plusieurs vers un, toutefois le client est le déclencheur du dialogue en demandant un service alors que le serveur attend passivement les requêtes,

Notons que dans certains cas, quand un client invoque un service, il peut transmettre une référence de type callbackà un objet, ce qui permet au serveur de le rappeler. Le client devient à son tour un serveur.

- ✓ **Transparence à la localisation :** Les processus client et serveur peuvent résider sur la même machine ou, par l'intermédiaire d'un réseau, sur deux machines différentes. Le logiciel Client/ Serveur masque aux clients la localisation du serveur en redirigeant les demandes de service si nécessaire. Un programme peut être client, serveur ou les deux.

✓ **Echange de message :** Clients et serveurs sont des systèmes à liaison épisodique qui interagissent au moyen de messages. Le message est un mécanisme d'émission de demandes de services et de réponses à celles-ci.

✓ **Encapsulation des services :** Le serveur est spécialiste, un message lui indique quel service est requis et c'est à lui de décider comment rendre ce service. Les serveurs peuvent être mis à niveau sans effet sur les clients tant que l'interface des messages reste la même.

✓ **Intégrité :** Le code et les données du serveur sont gérés de façon centralisée, chose qui garantit un moindre coût de maintenance et une meilleure intégrité des données tandis que les clients restent individuels et indépendants.

I.2.3. Intérêt du modèle Client/serveur

Le modèle Client/serveur s'impose dès que l'on souhaite décomposer l'exécution d'une application et faire en sorte que différentes machines (au sens large; matérielles et logicielles) y participent, et ce par opposition aux techniques de calcul centralisées sur mainframe; ou tout se fait au niveau du serveur central.

Les intérêts d'une telle décomposition sont multiples:

- Exploitation : il est alors facile d'utiliser (préparer) les entrées et comprendre les sorties d'un système ou d'un composant; ceci implique une meilleure maitrise du système d'information.

- Flexibilité : il est possible de modifier un système ou un composant pour le réutiliser dans des applications et environnements autres que ceux pour lesquels il a était conçu.

- Interopérabilité : les systèmes ou les composants peuvent communiquer; échanger les informations et les utiliser facilement.

-Mise à l'échelle : il est aisé de mettre à l'échelle le système ou le composant selon la dimension du problème à résoudre.

I.2.4. Principe de fonctionnement du client/serveur

Le Client/serveur est avant tout un mécanisme de dialogue entre deux processus. Ce modèle de communication est basé sur la fourniture de services par le processus serveur au processus client qui les demandent.

Le dialogue entre client et serveur consiste en l'envoi d'une requête au serveur qui l'exécute puis renvoie en retour la réponse appropriée (résultats) au client.

Le principe de fonctionnement du Client/serveur est illustré dans la figure suivante

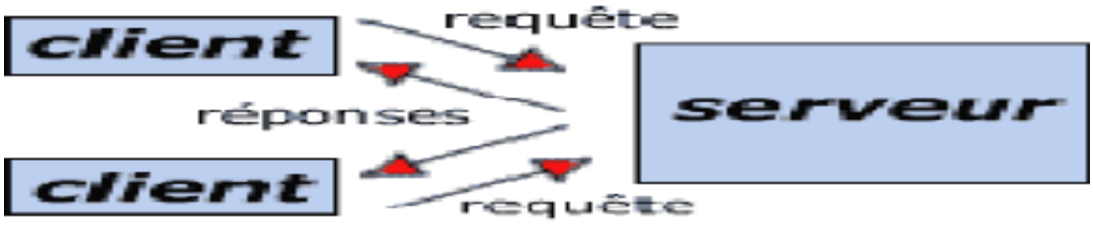


Figure 1.1. Principe de fonctionnement Client/serveur.

1.2.4.1. Concepts de base

Le vocabulaire de base de l'informatique Client/serveur manipule plusieurs concepts dont nous citons les notions de, Client, Serveur, Requête, Réponse, Middleware

✓ Client : processus demandant l'exécution d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.

✓ Serveur : processus accomplissant line opération sur demande d'un client et transmettant la réponse à ce client.

✓ Requête (requête) : message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.

✓ Réponse (reply): message transmis par le serveur à un client suite à l'exécution d'une opération contenant les paramètres de retour de l'application.

✓ Middleware: c'est le logiciel qui, situé au milieu, assure les dialogues entre les clients et les serveurs. Il est souvent hétérogène; en d'autres termes il constitue l'ensemble des services logiciels construits au-dessus du protocole de transport afin de permettre l'échange Requête-réponse de manière transparente en cachant l'hétérogénéité des composants mis enjeu (SGBD, réseau).

Notons au passage que le client est toujours le premier à engager la conversation en sa qualité de demandeur, ce par envoi d'une requête au serveur qui l'attend puis réalisé l'opération demandée et fournit la réponse au client.

Les appels aux services de transport mis en jeu sont au nombre de quatre:

- **SEND REQUEST ( ):** permet au client d'émettre le message décrivant la requête.

- **RECEIVE REQUEST ( ) :** permet au serveur de recevoir la requête en sa porte d'écoute.

- SEND REPLY ( ): permet au serveur d'envoyer la réponse sur la porte d'écoute du client.

- RECEIVE REPLY ( ): permet au client de recevoir la réponse en provenance du serveur.

La figure suivante illustre le dialogue Client/serveur

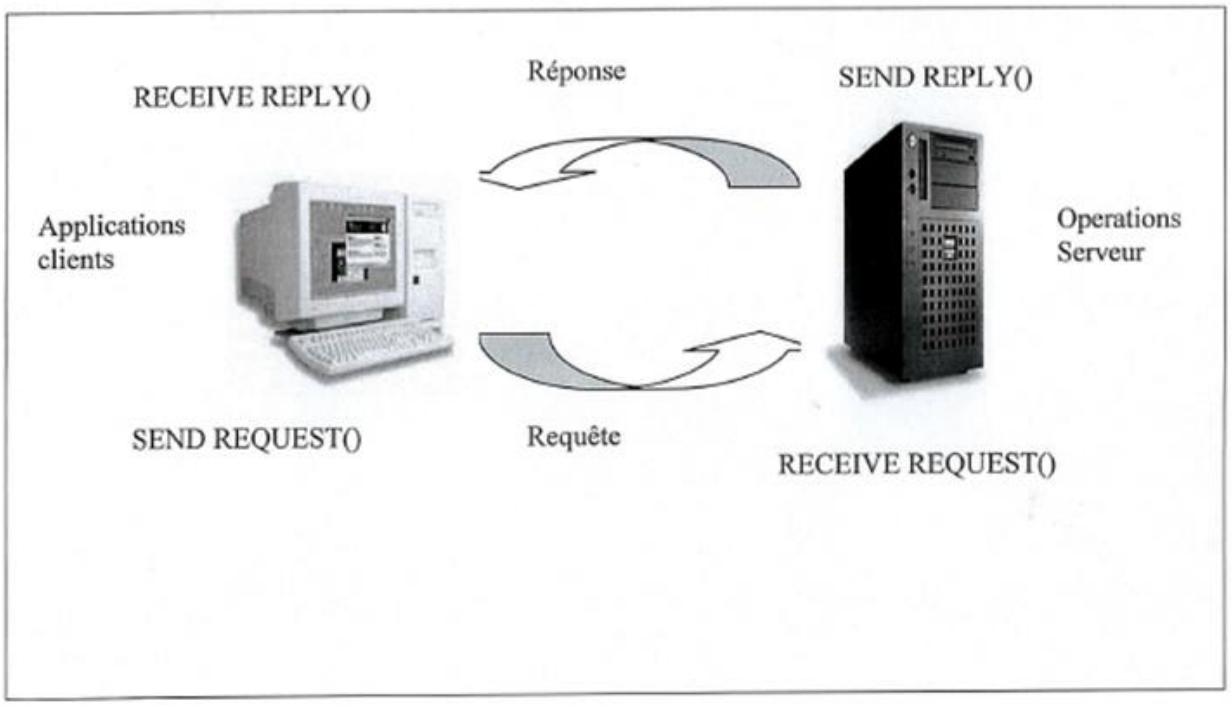


Figure 1.2. Le dialogue Client/serveur.

I.2.5. Modèles Client/serveur [4]

Alors que certains différencient les applications Client/serveur selon les services qu'elles offrent (selon les types de serveurs), et d'autres préfèrent les distinguer par la façon dont les fonctions distribuées se partagent entre le client et le serveur et utilisent plutôt les termes Orientation client, Orientation serveur; les prêtres du Client/serveur préfèrent utiliser les termes d'architectures Client/serveur à deux ou trois niveaux.

I.2.5.1. Types de serveurs

I.2.5.1.1. Serveur de fichiers

Dans le cas de serveur de fichiers, le client requiert des enregistrements de fichiers en émettant des requêtes au serveur de fichiers. Les serveurs de fichiers sont utiles pour partager des fichiers sur un réseau et ils sont indispensables pour créer des banques de documents

d'images, etc...Sa faiblesse réside dans l'obtention de l'information qui nécessite de nombreux échanges de messages sur le réseau.

I.2.5.1.2. Serveur de base de données

Dans le cas de serveur de base de données, le client émet des requêtes SQL sous forme de messages en direction du serveur. Le résultat de chaque requête est renvoyé au client. Les données ainsi que le code qui traite les requêtes, résident sur la même machine (serveur). Le serveur utilise sa propre capacité de traitement pour rechercher les données demandées au lieu de transmettre tous les articles au client et de le laisser en faire la sélection. Ainsi la puissance repartie est utilisée de façon beaucoup plus efficace.

I.2.5.1.3. Serveur de transactions

Dans ce modèle, les clients invoquent des procédures distantes résidant dans le serveur qui comporte un moteur de base de données SQL. Ces procédures distantes exécutent un ensemble d'instructions SQL. L'échange sur le réseau consiste en un seul message de Requête/Réponse (une réponse pour un bloc de requête SQL). Pour ce type de serveur l'application Client/serveur nécessite du code source au niveau du serveur.

I.2.5.1.4. Serveur de groupware

Le groupware s'intéresse à la gestion d'informations semi structurées telles que le texte, l'image, le courrier, la messagerie et l'ordonnancement des taches. Ces systèmes Client/serveur mettent les utilisateurs en contact direct les uns avec les autres, Microsoft Exchange est un exemple de ce type.

I.2.5.1.5. Serveur d'applications objet

Dans ce type de serveur, l'application Client/serveur est écrite sous forme d'un jeu d'objets communicants. Les objets clients communiquent avec les objets serveurs au moyen d'un courtier d'objet ou ORB (Object Request Broker). Le client invoque une méthode sur un objet distant, l'ORB localise une instance de la classe, appelle la méthode demandée et envoie les résultats à l'objet Client.

I.2.5.1.6. Serveur d'applications Web

L'internet est la plus grande application Client/serveur dite intergalactique. Ce nouveau modèle consiste en des clients légers et portables qui communiquent avec de très gros serveurs.

Le serveur web par exemple, renvoie des documents lorsque le client les demande par leurs noms. Clients et serveurs communiquent via un protocole de type RPC appelé HTTP (Hyper Text Transmission Protocol).

I.2.5.2. Orientation serveur & Orientation client

I.2.5.2.1. Orientation serveur

Le modèle orienté serveur place plus de fonctionnalité sur le serveur sur lequel tourne le gros de l'application. Les serveurs de groupeware, de transaction et les serveurs web sont des cas exemples des modèles orientés serveur.

I.2.5.2.2. Orientation client

La forme la plus traditionnelle est le modèle orienté client, dans lequel le gros de l'application tourne du coté client. Dans les modèles serveurs de bases de données et serveurs de fichiers, les clients connaissent l'organisation et la localisation des, données sur le serveur.

I.2.5.3. Architecture Client/serveur multi tiers [1]

I.2.5.3.1. Architecture Client/serveur à deux niveaux

L'architecture à deux niveaux (2-tiers) est l'architecture la plus classique, elle décrit les systèmes Client/serveur dans lesquels, la logique applicative est enfouie soit dans l'interface utilisateur chez le client, soit dans la base de données chez le serveur (ou dans les deux à la fois).

Dans cette architecture, le serveur exécute la requête du client et fournit directement le service, sans faire appel à d'autres intermédiaires,

L'architecture Client/serveur à deux niveaux est schématisée comme suit:

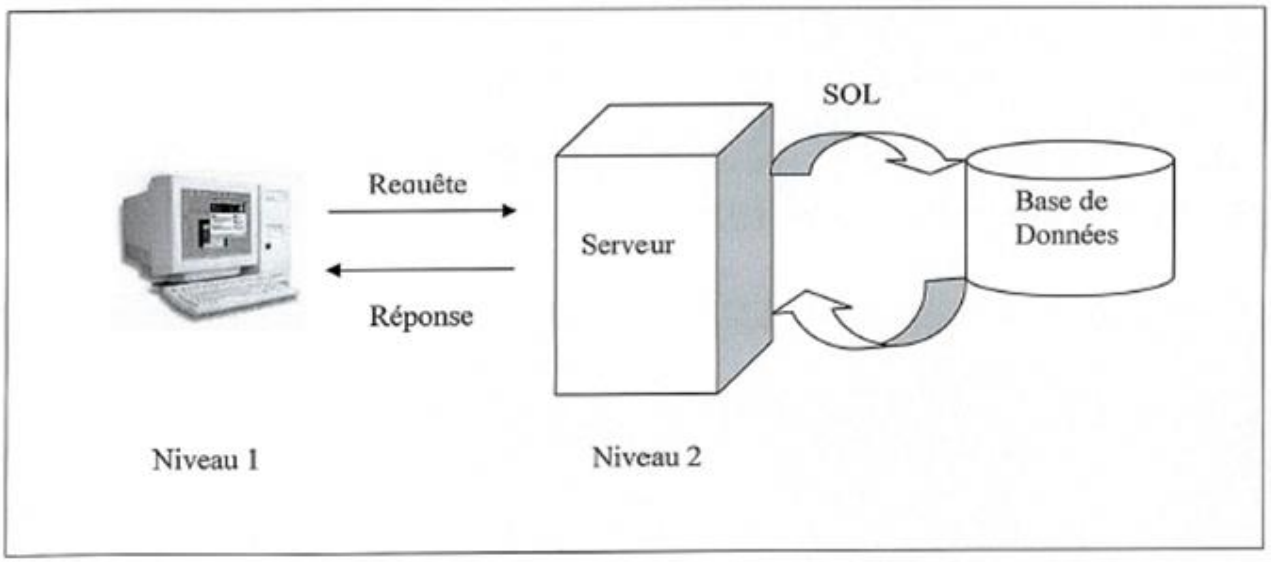


Figure 1.3. Architecture Client/serveur 2-tiers.

I.2.5.3.2. Architecture Client/serveur à trois niveaux

Dans cette architecture; la logique applicative réside dans un niveau intermédiaire, séparément des données et de l'interface utilisateur. Les trois niveaux de cette architecture sont:

-Le client (niveau 1) : demandeur de ressources.

- Le serveur d'application (niveau 2) : appelé aussi middleware, il est chargé de fournir les ressources mais en faisant appel à un autre serveur.

- Le serveur de base de données (niveau 3) :c'est celui qui fournit le service auserveur d'application.

La figure ci-jointe montre les niveaux de l'architecture 3-tiers :

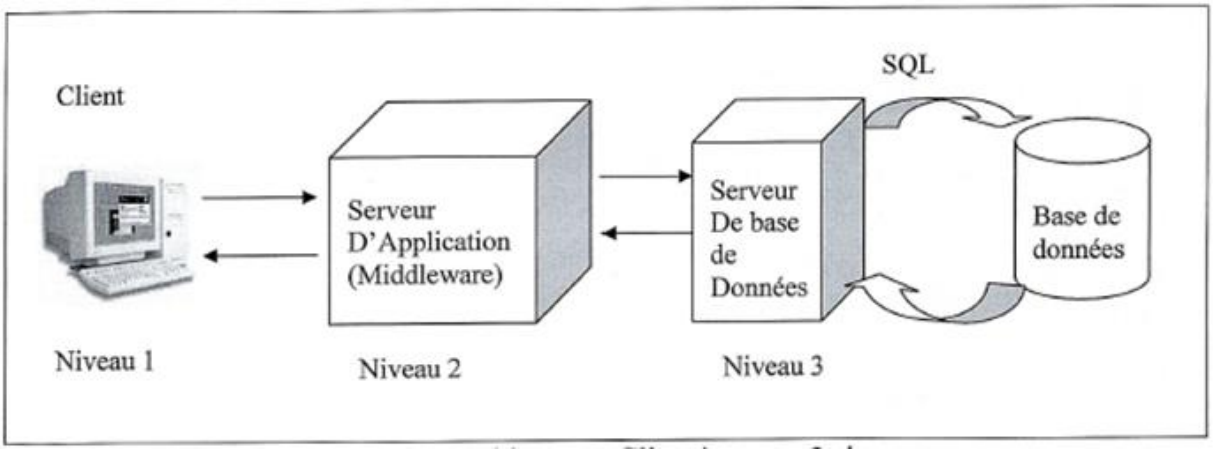


Figure 1.4. Architecture Client/serveur 3-tiers.

En théorie, les systèmes à trois niveaux sont, contrairement à ceux de deux niveaux, mieux dimension nables, plus robustes et plus souples.

Le modèle à trois niveaux est recommandé si l'application possède une des caractéristiques suivantes :

- Nombreux services ou nombreuses classes (plus de 50).
- Application écrite dans plusieurs langages ou par différentes organisations.
- Sources de données multiples et hétérogènes.
- Application évolutive (modifications et ajouts), de longévité supérieure à 3ans.
- Importante charge de traitement; plus de 50 000 transactions par jour ou plus de 300 utilisateurs accédant simultanément à la même base de données,
- Importante communication inter applications.
- Croissance de l'application susceptible d'amener l'une des conditions ci-dessus.

Le niveau intermédiaire de la plupart des applications à trois niveaux n'est pas implémenté comme un programme monolithique, mais d'une façon modulaire. Chaque composant se charge d'une fonction spécifique et relativement petite. Un composant peut en appeler un autre pour satisfaire une requête du client. Ainsi, le modèle à trois niveaux est en fait à N niveaux.

Cependant, il faut noter que la vulnérabilité du système augmente en fonction du nombre de niveaux; remarquons qu'en architecture 2-tiers le serveur de données est le maillon faible au sein du système, étant donné que toute l'application est architecturée autour de lui. En architecture 3-tiers l'arrêt du Middleware induit celui de tout le système; cela s'ajoute au fait que le serveur de données est toujours maillon faible comme en architecture 2-tiers.

I.2.5.3.3. Principales fonctions du middleware

- Procédures d'établissement de connexion.
- Récupération des résultats
- Procédure de fermeture de connexion.
- Conversion de données organisées (tables structurées, objets reliés entre eux du fait de l'hétérogénéité des plates-formes).
- Exécution des requêtes.
- Gestion des accès concurrents.
- Accès aux données à distance.
- Initialisation des processus.
- Terminaison des processus.

I.2.5.3.5. Architecture Client/serveur multi niveaux

Dans l'architecture à trois niveaux, chaque serveur (niveau 1 et 2) effectue une tache (un serveur) spécialisée. Ainsi, un serveur peut utiliser le service d'un ou plusieurs autres serveurs afin de fournir son propre service par conséquence, l'architecture à trois niveaux est potentiellement une architecture à N niveaux

Le schéma suivant illustre ce type d'architecture

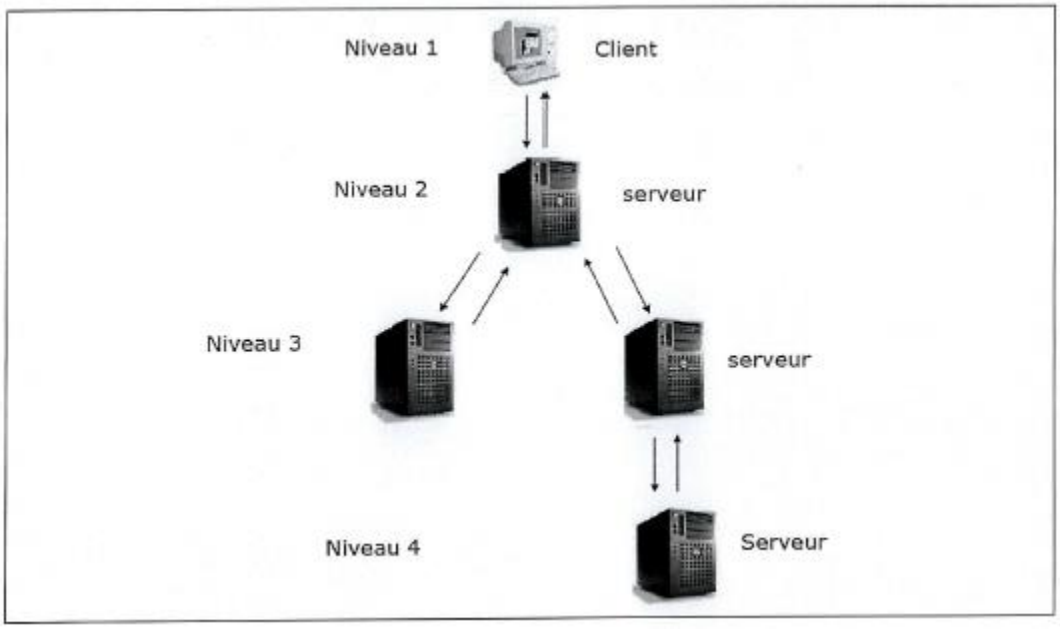


Figure 1.5. Architecture Client/serveur multi niveaux.

I.2.6. Avantages et inconvénients de l'architecture client/ serveur

I.2.6.1. Avantages

Le modèle client/serveur s'impose dès que l'on souhaite composer l'exécution d'une application et faire en sorte que différentes machines (matérielles et logicielles) y participant et ce par opposition aux techniques de calcul centralisés sur mainframe, tout se fait au niveau du serveur central.

Le modèle client/serveur est particulièrement recommandé pour des réseaux qui nécessitent un grand niveau de fiabilité, ses principaux avantages sont :

- Des ressources centralisées : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisées afin d'éviter les problèmes de redondances et de contradiction.

- Une meilleure sécurité : car le nombre de points d'entrée permettant l'accès aux données est moins important.

- Une administration au niveau serveur: les clients ont peu d'importances dans ce modèle, ils ont moins besoins d'être administrés.

- Un réseau évolutif grâce à cette architecture, on peut supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures.

I.2.6.2. Inconvénients

L'architecture client/serveur à tout de même quelques lacunes parmi lesquelles:

- Un cout élève : dû à la technicité du serveur.

- Un maillon faible: le serveur est seul maillon faible du réseau, étant donné que tout le réseau est construit autour de lui.

I.3. Discussion

Le modèle client /serveur est la base de tous les services réseaux informatiques, c'est pour cela que nous nous sommes intéressés à l'étude de ce modèle. Le but de ce chapitre est de présenter les différentes notions de base de ce modèle comme le middleware, les protocoles, les sockets et l'appel de procédure à distance.

Nous concluons que le modèle Client/serveur est l'un des divers modèles d'informatique en réseau visant l'utilisation de multiples ordinateurs dans un même système. Le réseau utilisé pour le développement de notre application est configuré en architecture Client/serveur, le serveur est configuré sous le système d'exploitation Windows, les machines se connectent via le protocole de transmission de données TCP/IP, la majorité des traitements s'effectuent au niveau du client et les résultats sont renvoyés au serveur ce qui permet de dire que le réseau est orienté vers le client.

II.1. Préambule

Il fut un temps où remplacer chaque lettre d'un message par celle située trois positions plus loin dans l'alphabet suffisait aux empereurs romains pour se mettre à l'abri des regards indiscrets. Mais, depuis que les curieux ont appris à lire, à compter jusqu'à trois puis jusqu'à vingt-six et à utiliser un ordinateur, les secrets de Jules César se sont passablement éventés. Face aux progrès technologiques, ses successeurs durent alors recourir à l'ingéniosité des cryptographes. Nourris par les mathématiques discrètes, la théorie de la complexité et la théorie de l'information, leur imagination est toujours à la recherche de techniques efficaces et infaillibles permettant de protéger les données confidentielles d´éventuels indiscrets.

L'avènement de la cryptographie moderne, depuis la découverte en 1976 des systèmes à clé publique, n'a pas apporté de solution parfaite aux cryptographes dans leur quête effrénée de systèmes toujours plus rapides et plus surs.

Les solutions apportées par la cryptographie pour une meilleure sécurité de l'information se répartissent en deux grandes catégories suivant leur fonctionnalité : les *algorithmes de chiffrement*, qui assurent la confidentialité des données, et les *algorithmes d'authentification*, qui garantissent en l'authenticité et la provenance.

II.2. Cryptologie

La cryptologie est la science du secret. Elle se divise en deux disciplines :

- La cryptographie qui est l'étude des algorithmes permettant la protection d'informations (numériques). Ces algorithmes sont appelés cryptosystèmes ;
- La cryptanalyse qui est l'étude du niveau de sécurité des cryptosystèmes fournis par les cryptographes. **[5]**

II.3. Cryptographie

La cryptographie est l'art de rendre inintelligible, de crypter, de coder un message à ceux qui ne sont pas habilités à en prendre connaissance ou est un ensemble des principes, méthodes et techniques dont l'application assure le « crypter » et le « décrypter » des données **[6,7]** (Figure II.1).

La signification de « crypter » et « décrypter » est donnée successivement comme suit :

- **Crypter** : brouiller l'information, la rendre "incompréhensible"
- **Décrypter** : rendre le message compréhensible

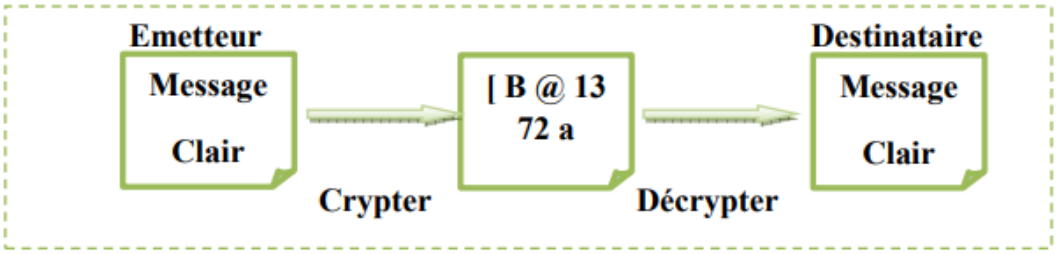


Figure II.1. Chiffrement / déchiffrement.

II.4. Objectifs de la cryptographie

Les principaux objectifs à garantir par l'application de la cryptographie sont :

a. **Confidentialité** : un mécanisme pour transmettre des données de telle sorte que seul le destinataire autorisé puisse les lire.

b. **Intégrité des données** : un mécanisme pour s'assurer que les données reçues n'ont pas été modifiées durant la transmission, frauduleusement ou accidentellement.

c. **Authentification :** un mécanisme pour permettre d'authentifier les utilisateurs de façon à limiter l'accès aux données, serveurs et ressources aux seules personnes autorisées (un mot de passe par un nom de login ou un certificat numérique).

d. **Non-répudiation** : un mécanisme pour enregistrer un acte ou un engagement d'une personne ou d'une entité de telle sorte que celle-ci ne puisse pas nier avoir accompli cet acte ou pris cet engagement. Ce mécanisme se décompose :

- Non-répudiation d'origine l'émetteur ne peut nier avoir écrit le message.
- Non-répudiation de réception le receveur ne peut nier avoir reçu le message.
- Non-répudiation de transmission l'émetteur du message ne peut nier avoir envoyé le message**. [8]**

II.5. Techniques de cryptographie

Des algorithmes basés sur des clés sont utilisés pour assurer les objectifs de la cryptographie. Ces algorithmes sont définis par plusieurs types de cryptographie ou crypto systèmes.

II.6. Définition de la clé

On appelle clé une valeur utilisée dans un algorithme de cryptographie, afin de chiffrer une donnée. Il s'agit en fait d'un nombre complexe dont la taille se mesure en bits. On peut imaginer que la valeur correspondant à 1024 bits est absolument gigantesque. Voir aussi bits

bytes. Plus la clé est grande, plus elle contribue à élever la sécurité à la solution. Toutefois, c'est la combinaison d'algorithme complexe et de clés importantes qui seront la garantie d'une solution bien sécurisée.

Les clés doivent être stockées de manière sécurisée et de manière à ce que seul leur propriétaire soit en mesure de les atteindre et de les utiliser. **[9]**

II.7. Cryptographie symétrique

Historiquement, le concept de cryptographie symétrique est le premier à avoir émergé, permettant notamment d'assurer la sécurité des communications entre deux interlocuteurs. En cryptographie symétrique, une même clé sert à réaliser les opérations de chiffrement et de déchiffrement (Figure II.2). Cette clé doit être tenue secrète pour assurer la sécurité de la communication, c'est pourquoi on désigne aussi ce domaine par cryptographie à clé secrète. Le premier algorithme de cette famille, le plus connu, est sans doute le chiffre de César. Son principe consiste à appliquer, à chaque symbole du texte clair, une rotation de valeur fixe dans l'alphabet de référence. Dans ce cas, c'est la valeur de la rotation qui fait office de clé secrète.

De par sa construction, la cryptographie symétrique souffre de deux problèmes majeurs dans son application. Le premier concerne l'échange préalable de la clé secrète entre les deux interlocuteurs.

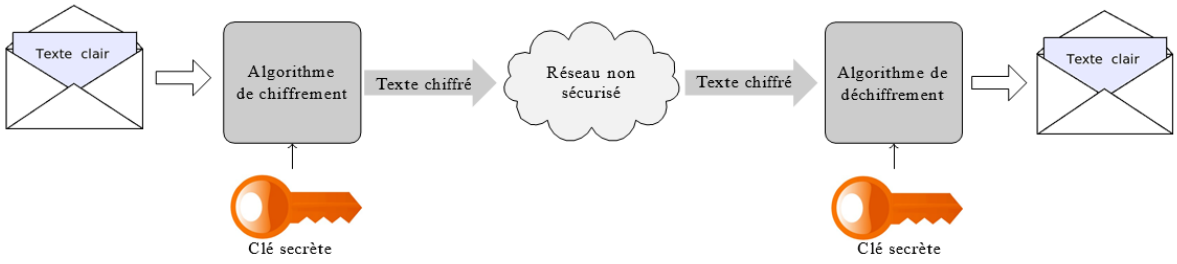


Figure II.2. Schéma de cryptographie symétrique

En effet, pour garantir la sécurité des communications futures, il faut que cette clé ne soit connue par personne d'autre. Cela nécessite de sécuriser un échange préalable de cette clé secrété. L'autre problème relatif à la cryptographie symétrique est la gestion de ces clés. Puisqu'une clé secrète ne permet de sécuriser qu'une seule communication entre deux individus, pour un réseau de n individus, il faudra alors gérer $n(n-1)/2$ clés. En conséquence, le nombre de clés à gérer évolue comme le carré du nombre d'individus sur le réseau. Ces deux problèmes ont très certainement motivé l'invention de la cryptographie asymétrique à la fin des années 1970.

Classiquement, on distingue deux types d'algorithmes de chiffrement symétrique : les algorithmes de chiffrement par blocs et les algorithmes de chiffrement à flot. Comme leur nom le suggère, les algorithmes de chiffrement par blocs prennent en entrée des blocs de texte clair de taille fixe (typiquement 64 ou 128 bits) et retournent des chiffres de même longueur. Cette catégorie d'algorithme de chiffrement symétrique est sans doute la plus utilisée en pratique avec, comme représentants, le DES 3 **[10]** et le standard actuel AES 4 **[11]**. Les algorithmes de chiffrement à flot, quant à eux, permettent de réaliser un chiffrement bit à bit d'un flot de données en entrée. La clé secrète permet d'initialiser la génération d'une longue séquence de bits, la suite chiffrante, à la manière d'un générateur pseudo aléatoire. Cette suite chiffrante est ensuite utilisée pour masquer les bits de texte clair suivant le principe du chiffrement à masque.

II.7.1. Chiffrement par bloc (Block Cipher)

C'est une des deux grandes catégories de chiffrements modernes en cryptographie symétrique. Il consiste à un découpage des données en blocs de taille généralement fixe (souvent une puissance de deux comprise entre 32 et 512 bits). Les blocs sont ensuite chiffrés les uns après les autres. Il est possible de transformer un chiffrement de bloc en un chiffrement par flot en utilisant un mode d'opération comme ECB (chaque bloc chiffré indépendamment des autres) ou CFB (on chaîne le chiffrement en effectuant un XOR entre les résultats successifs).

Un exemple de taille de bloc et de clé utilisés par les algorithmes les plus connus :

- DES : blocs de 64 bits, clé de 56 bits.
- IDEA : blocs de 64 bits, clé de 128 bits.
- AES : blocs de 128 bits, clé de 128 à 256 bits.

Pour cette catégorie, nous allons présenter deux algorithmes très connus DES et AES en mettant l'accent sur l'AES car il est devenu le standard recommandé pour le chiffrement symétrique **[12]**.

II.7.1.1. DES (Data EncryptionSandard)

DES a été développé à la fin des années 1970s comme un standard du gouvernement US pour protéger l'information sensible. Le DES est officiellement défini dans la publication FIPS 46-3 et il est public. DES est encore supporté dans des outils de cryptographie pour les O.S des dispositifs mobiles et les cartes à puce. C'est un chiffrement qui transforme des blocs de 64 bits avec une clé secrète de 56 bits. Il a une conception basée sur le schéma de Feistel au moyen de

permutations et de substitutions. L'attaque de force brute est possible sur une clé DES. Cette attaque a été réalisée par DES Cracker de EFF-ElectronicFrontier Fondation en juin 1998 (Elle a trouvé une clé DES dans moins de 3 jours).

Avec la technologie actuelle (des CPU très rapides et moins chers), il est hautement possible de cracker DES. La solution a été au premier temps d'adopter le triple DES (TDES ou 3DES) : 3 applications de DES à la suite avec 2 clés différentes (112 bits) (Figure 3.3).
TDES utilise une taille de blocs est 64 bits et de clé comprise entre 128 bits et 192 bits.

Figure II.3. Algorithme TDES

TDES est suffisant en sécurité mais il est trois fois plus lent que DES. Un nouvel algorithme remplace DES, c'est l'AES (Advanced Encryption Standard). **[13]**

II.7.1.2. AES (Advanced Encryption Standard)

En 1997, le NIST (National Institue of Standards and Technology) a lancé un appel d'offre pour un algorithme de chiffrement symétrique avec un bloc de taille 128 bits et supporte des clés de 128, 192 et 256 bits. Les critères d'évaluation de l'offre comprenaient la sécurité, la puissance de calcul, les contraintes de la mémoire des petits dispositifs (comme la carte à puce), la plateforme logicielle et matérielle et la flexibilité. Un total de 15 algorithmes ont été envoyés et 5 ont été sélectionné parmi ces 15 algorithmes **[14]**.

L'algorithme de Rijndael a été sélectionné pour devenir l'AES en 2001. Rijndael a été conçu par Joan Daemen et Vincent Rijmen, deux chercheurs de la Belgique. Les autres algorithmes sont : Serpent, Twofish, RC6, et MARS. L'AES n'utilise pas le schéma de Feistel mais il utilise des opérations mathématiques comme des substitutions, des permutations et des XORs. Il a plusieurs rounds identiques (de 10 à 14) et leur nombre dépend de la taille de la clé. L'AES opère au niveau octet ce qui permet une implémentation efficace au niveau matérielle et logicielle. L'AES est un standard, donc libre d'utilisation, sans restriction d'usage ni brevet. NIST spécifie actuellement AES dans le document FIPS 197, comme le nouveau standard de chiffrement symétrique. AES est approuvé pour utilisation par les organisations du gouvernement U.S pour protéger l'information sensible non classifiée.

AES a trois niveaux forts de sécurité : 128 bits, 192 bits et 256 bits. La sécurité 128 bits fournira au moins 30 ans de protection. AES ne fournit pas seulement une sécurité supérieure à TDES mais il délivre aussi une meilleure performance. Une meilleure sécurité et une

meilleure performance rendent l'AES une alternative plus attractive que TDES et un bon choix pour un algorithme de chiffrement symétrique **[14]**.

AES est également un candidat particulièrement approprié pour les dispositifs mobiles limités en ressources de calcul et de stockage. Le monde de la 3G (3ème génération de dispositifs mobiles) a adopté l'algorithme AES pour son schéma d'authentification.

II.7.1.3. Autres finalistes d'AES

Durant la compétition avec AES, Serpent et Twofish avaient une performance faible comparée à Rijndael. RC6 et MARS ont des problèmes de sécurité et d'efficacité. RC6 a été cassé à au moins de 17 rounds sur 20 pendant la compétition avec AES. MARS était plus couteux en calcul pour l'implémenter comparé aux autres finalistes d'AES **[15].**

II.7.2. Chiffrements de flux (Stream Cipher)

Les algorithmes de chiffrement de flux peuvent être définis comme étant des algorithmes de chiffrement par blocs, où le bloc a une dimension unitaire (1 bit, 1 octet, etc.) ou relativement petite. Leurs avantages principaux viennent du fait que la transformation (méthode de chiffrement) peut être changée à chaque symbole du texte clair et du fait qu'ils soient extrêmement rapides. De plus, ils sont utiles dans un environnement où les erreurs sont fréquentes car ils ont l'avantage de ne pas propager les erreurs (diffusion). Ils sont aussi utilisés lorsque l'information ne peut être traitée qu'avec de petites quantités de symboles à la fois **[16]**. Quelques algorithmes de cryptographie symétrique par flot :

- A5 : utilisé dans les téléphones mobiles de type GSM pour chiffrer la communication par radio entre le mobile et l'antenne-relais la plus proche.
- RC4, le plus répandu, conçu par Ronald Rivest, utilisé notamment par le protocole WEP, un algorithme récent de Eli Biham – E0 utilisé par le protocole Bluetooth.

II.8. Cryptographie asymétrique

Le concept de cryptographie asymétrique, ou cryptographie à clé publique a été introduit en 1976 par WhitfieldDiffie et Martin Hellman dans l'article fondateur **[17]**. A l'époque, les auteurs ont présenté le concept sans toutefois en proposer d'instance concrète. Il a fallu attendre l'invention du RSA **[18]** et du protocole d'échange de clé Diffie-Hellman**[19]** pour voir les premières applications de cryptographie asymétrique.

Contrairement à la cryptographie symétrique, ce nouveau système repose sur l'utilisation de deux clés distinctes servant à réaliser respectivement les opérations de chiffrement et de

déchiffrement. Classiquement, le destinataire choisit la clé de déchiffrement, la clé privée, dont il dissimule la valeur. En utilisant une fonction à trappe et la clé privée, il calcule ensuite la clé qui sera utilisée pour le chiffrement : la clé publique. L'utilisation de fonctions à trappe est particulièrement intéressante ici car ces fonctions sont faciles à calculer mais difficiles à inverser si l'on ne connaît pas la trappe. Aussi, en cryptographie asymétrique, c'est la clé privée qui fait office de trappe. La clé publique est ensuite diffusée ce qui permet aux autres individus d'envoyer des messages chiffrés que seul le destinataire légitime (i.e. Le détenteur de la clé privée associée) pourra déchiffrer (Figure II.4). Ainsi, dans le cadre de la cryptographie asymétrique, il n'est pas nécessaire d'échanger préalablement un secret pour communiquer de manière sécurisée.

Toutefois, l'utilisation de clés publiques peut poser des problèmes. En effet, en l'état, il est difficile d'identifier réellement la personne qui diffuse sa clé publique. Il se pourrait qu'un attaquant diffuse une clé publique en usurpant l'identité d'un individu. Il serait alors capable de déchiffrer les messages adressés à la victime. Ce problème peut être réglé par le déploiement d'infrastructures de gestion de clés publiques.

En pratique, les méthodes de chiffrement asymétrique sont nettement plus lentes que celles de chiffrement symétrique. Cela s'explique notamment par la complexité des opérations mathématiques qu'elles doivent réaliser ainsi que par la longueur des clés recommandées pour garantir un niveau de sécurité acceptable (par exemple, la recommandation minimale de l'ANSSI 12 pour la taille des clés RSA est de 2048 bits **[20]**). Par conséquent, les méthodes de chiffrement asymétrique ne sont pas intéressantes pour sécuriser des canaux de communications sur lesquels transite un important volume de données. En revanche, elles sont très pratiques pour sécuriser l'échange d'une clé de chiffrement symétrique qui permettra ensuite d'utiliser un algorithme de chiffrement par blocs, plus rapide, pour échanger des données.

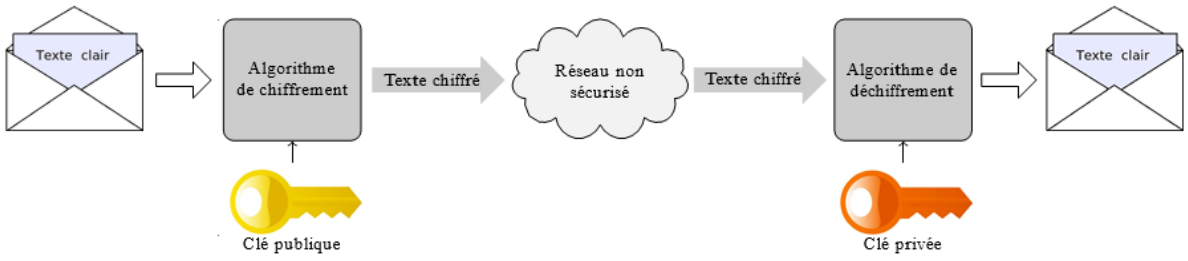


Figure II.4. Schéma de cryptographie asymétrique

L'apparition de la cryptographie asymétrique a permis l'émergence du concept de signature électronique. Comme pour une signature manuscrite apposée au bas d'un document, sa version électronique peut garantir les deux propriétés cryptographiques que sont l'authentification (identifier l'individu qui a signé le document) et l'intégrité (vérifier que le document n'a pas été modifié après sa signature). La figure II.5 décrit un schéma général de signature numérique. L'utilisation de la clé privée pour le calcul de la signature permet aux autres individus, connaissant la clé publique associée et le document original, de vérifier facilement sa validité. La validité de la signature prouve alors que c'est bien le possesseur de la clé privée qui a signé le document et que le document n'a pas été modifié après sa signature. Ainsi, la signature électronique présente bien les propriétés cryptographiques attendues.

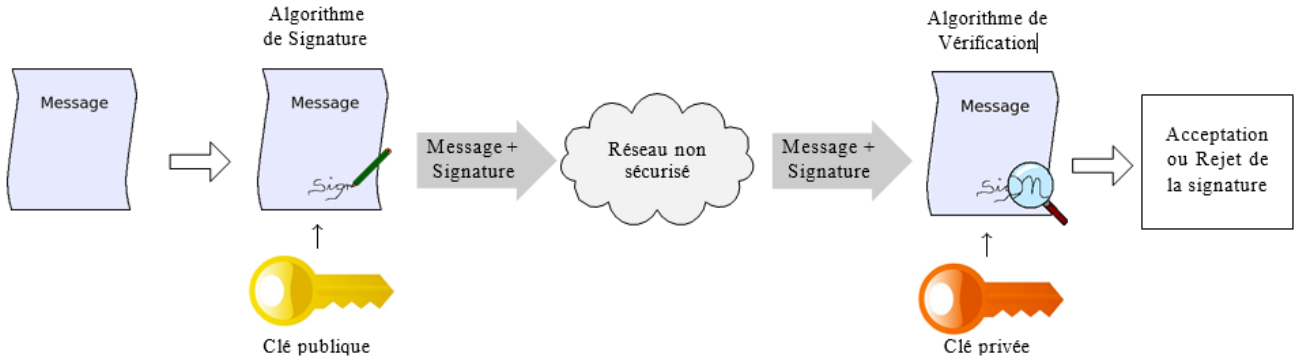


Figure II.5. Schéma de signature électronique

Dans la partie suivante nous présenterons l'un des algorithmes les plus utilisés en cryptographie asymétrique à savoir l'algorithme RSA que l'on utilisera pour ce travail de fin d'étude.

II.8.1. Cryptage RSA

Le cryptage RSA, du nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adleman, est le premier algorithme de chiffrement asymétrique. Il a été découvert en 1977 au Massachusetts Institute of Technology. Le principe de ce cryptage est d'utiliser une clé publique pour crypter les données et une clé privée qui servira à les décrypter **[21]**. Nous allons maintenant décrire les opérations qu'on peut réaliser avec des algorithmes :

- **Génération des clés**

Le RSA fonctionne à partir de deux nombres premiers. Ces deux nombres doivent être très grands et calculer la clé privée et la clé publique comme dans l'algorithme suivant :

Algorithme de Génération des clés

Entrées : Deux nombres premiers p et q.

Sorties : Une clé privée d et une clé publique e.

Prendre un nombre aléatoire p et q.

Si p = q alors Prendre un nombre aléatoire p et q.

Sinon $N = p \times q$.

Fin Si

Calcul $\Theta(N) = (p - 1)(q - 1)$.

Prendre un nombre aléatoire e dans l'intervalle

$[1, \Theta(N)]$. Si $PGCD(e, \Theta(N)) \neq 1$ alors

Prendre un nombre aléatoire e dans l'intervalle $[1, \Theta(N)]$.

Si non Calcule $d \equiv e^{-1} \bmod(\Theta(N))$.

Fin Si

- **Chiffrement du message**

Supposons maintenant que les intervenants A et B possèdent chacun son module RSA et ses clés N_A, e_A ,d_A pour A et N_B, e_B, d_B pour B. Si A veut envoyer un message M à B, il peut procéder comme dans l'algorithme suivant :

Algorithme de Chiffrement d'un message

Entrées : Un message clair M et la clé publique
(N_B, e_B). Sorties : Un message chiffré C.

Transformer le message en un nombre entier M de l'intervalle $[0, N_B - 1]$

Calculer $C \equiv M^{eB} \bmod N_B$ et

$C < \Theta N_B$. Envoyer le message C.

- **Déchiffrement du message**

B a reçu un message chiffré C de la part de A. Alors B peut le déchiffrer en utilisant sa clé secrète d_B comme dans l'algorithme suivant :

Algorithme de Déchiffrement d'un message

Entrées : Un message chiffré C et la clé privée
(N_B,d_B). Sorties : Un message clair M.

Calculer $M = C^{d}B \mod N_B$

Retourner le message M.

II.8.2. Considérations de sécurité pour l'utilisation de RSA

- Il est fondamental d'utiliser CRT (ChineseRemainderTheorem) dans RSA pour augmenter la vitesse de déchiffrement.
- Il est aussi fondamental d'utiliser le schéma de cryptage RSAES-OAEP au lieu de RSA seul pour les opérations de chiffrement et déchiffrement.
- Il est fondamental d'utiliser une clé RSA 2048-bit comme un minimum pour l'utilisation de la cryptographie RSA dans les nouvelles applications (à partir de 2010).

II.9. Discussion

La cryptologie est la science des messages secrets. Longtemps restreinte aux usages diplomatiques et militaires, elle est maintenant une discipline scientifique à part entière, dont l'objet est l'étude des méthodes permettant d'assurer les services d'intégrité, d'authenticité et de confidentialité dans les systèmes d'information et de communication. Cette science est devenue indispensable pour palier et contrecarrer le problème de cyberattaque dans les différentes entreprises.

Dans ce chapitre, nous avons présenté les différentes notions de base sur la cryptologie et ses grandes catégories. Nous avons également abordé les objectifs et les techniques de la mise en place de cette science désormais indispensable pour la sécurité des entreprise.

III.1. Préambule

Dans le monde de l'informatique, la sécurité est un point crucial pour toute organisation qui échange des données numériques sur un réseau (interne ou externe). Lorsqu'il s'agit d'échanger des données confidentielles ou personnelles, il peut toujours subsister un doute sur l'interlocuteur avec lequel on communique, ou la crainte que les données transmises ne soient interceptées et exploitées à des fins mal intentionnées. Les infrastructures à clés publiques, dites **PKI** (*Public Key Infrastructure*), ont été conçues pour bâtir une relation de confiance dans un environnement parfois mal sécurisé.

III.2. Infrastructure à clé publique

III.2.1. Présentation des PKI

Les infrastructures à clés publiques (**PKI**) ont été créées dans le but de prouver aux autres que vous êtes vraiment celui que vous prétendez être lors d'un échange de données à travers un réseau informatique.

L'identité de chaque entité, composant du réseau, individu, système, ou autres peut être vérifiée à l'aide d'un certificat issu d'une autorité de certification approuvée. Les certificats numériques font partie intégrante d'une infrastructure à clés publiques et peuvent être totalement transparents pour un utilisateur utilisant quotidiennement un ordinateur. On retrouve l'utilisation des certificats dans toutes sortes d'éléments, que ce soit pour sécuriser une connexion VPN IPsec (*Virtual Private Network InternetProtocol security*), un échange d'e-mails, effectuer un paiement en ligne sur un site commercial (site web en HTTPS utilisant un certificat SSL) ou sécuriser l'accès à des fichiers ou répertoires en utilisant un système de chiffrement EFS (*Encrypting File System*). Les infrastructures à clés publiques nous aident donc à sécuriser les communications informatiques et notamment à établir des liens de confiances entre des personnes physiques et des certificats numériques tels que le ferait une carte nationale d'identité. Un administrateur en charge d'une infrastructure à clés publiques se doit de distribuer avec une extrême vigilance les certificats numériques aux personnes membres de son organisation et révoquer s'il le faut les certificats des utilisateurs ne faisant plus partie de l'entreprise.

Par exemple, lorsque vous consultez le site Internet de votre banque pour vérifier vos comptes, l'accès y est probablement sécurisé à l'aide d'une page web en HTTPS, utilisant un certificat SSL. Si vous affichez les propriétés de ce certificat, vous pouvez constater que ce dernier possède les informations de l'autorité de certification ayant délivré le certificat pour votre banque. Votre navigateur accepte d'afficher la page demandée car le certificat a été délivré

par une autorité de certification commerciale approuvée telle que VeriSign. Par défaut, votre navigateur Internet contient une liste d'autorités de certification commerciales approuvées. Cette liste est mise à jour automatiquement lorsque vous effectuez une mise à jour de votre système d'exploitation. La liste des certifications commerciales approuvées est disponible dans les options de notre navigateur web :

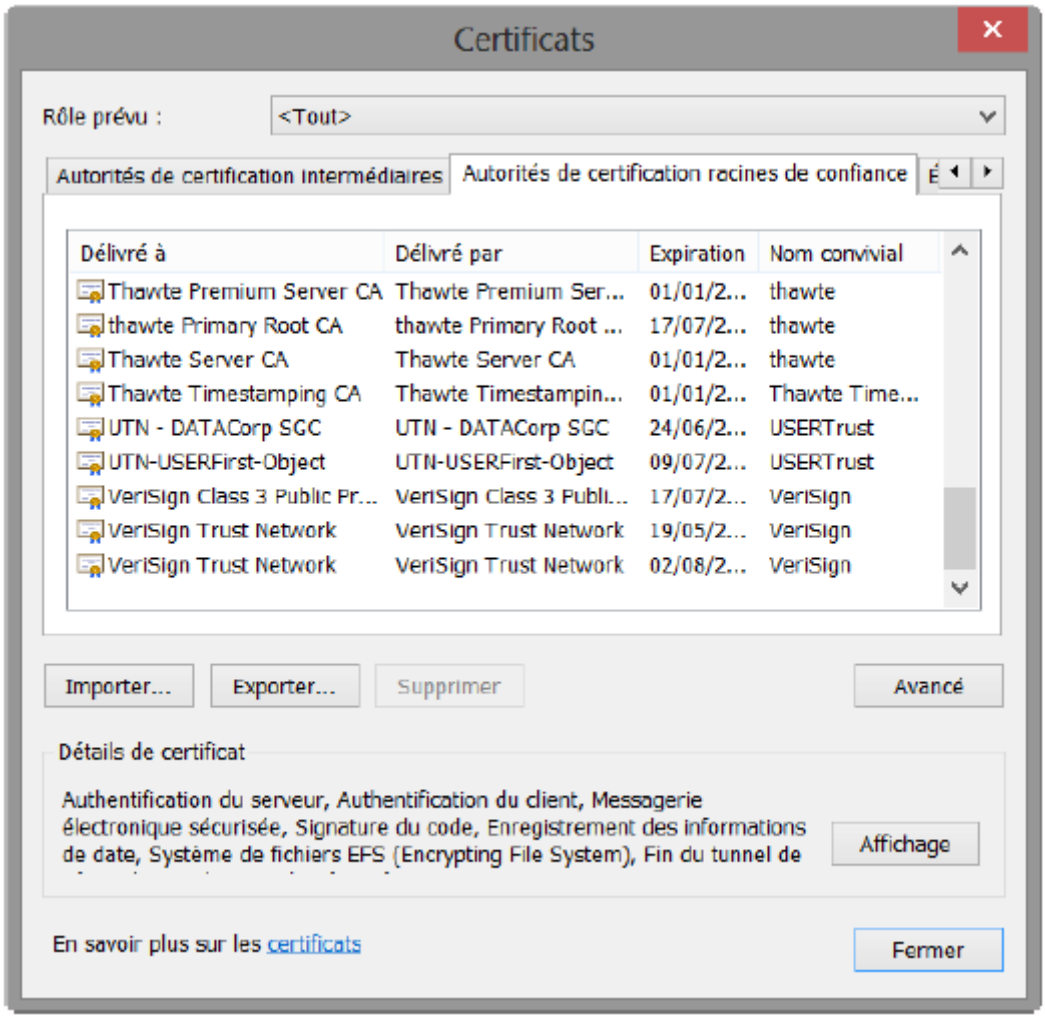


Figure III.1. Liste des certificats approuvés.

III.2.2. Composants d'une PKI

Une infrastructure à clés publiques est composée de plusieurs éléments dont certains offrent un service aux utilisateurs. On retrouve par exemple les éléments suivants :

- **Autorités de certification** (CA : *CertificateAuthority*) : permettent de créer et de gérer des certificats.

Les autorités de certification peuvent être structurées sous différentes formes de hiérarchie que l'on peut définir de hiérarchies à deux ou trois niveaux.

- **Certificats numériques** : les certificats sont distribués par une autorité de certification approuvée par l'organisation (CA interne ou commerciale). Ces certificats servent à authentifier un service, un serveur, un utilisateur ou encore un site commercial. Dans le cas d'un site web par exemple, le certificat délivré doit posséder, comme attribut, le nom DNS du site web accédé par les utilisateurs via une URL. Si le nom ne correspond pas, l'utilisateur ne pourra accéder à l'information sécurisée.

Cela permet d'attribuer un certificat valide pour un unique service. Idem concernant un certificat utilisateur contenant le nom et prénom de l'intéressé. L'authentification par certificat ne fonctionne que si l'identité de l'utilisateur est vérifiée.

- **Modèles de certificats** : les modèles de certificat permettent à des administrateurs de générer des certificats préconfigurés en fonction de l'usage souhaité. Ces certificats sont également personnalisables en fonction du besoin.

- **Clés publiques et des clés privées** : utilisées pour le chiffrement et déchiffrement des données.

-**Postes clients** : les utilisateurs peuvent utiliser leur poste client pour demander des certifications à une autorité de certification déclarée dans l'Active directory ou via le formulaire d'une page web de l'autorité de certification. Ils peuvent également accéder à des sites sécurisés par des certificats. Les postes clients possèdent une liste d'autorités de certification approuvées (accessible depuis un navigateur web ou le composant logiciel enfichable *Certificat*), leur permettant de valider l'accès à des sites web utilisant des certificats délivrés par des autorités de certification commerciales.

- **Listes de révocation de certificats** (CRL : *CertificateRevocationLists*) : représentent des listes publiées par les autorités de certification et contenant les certificats numériques invalides ou révoqués.

Une autorité de certification maintient à jour cette liste contenant le détail de l'attribution des certificats en indiquant les dates de validités ainsi que les certificats ayant été révoqués.

Lorsqu'un certificat est présenté pour l'accès à un service, une ouverture de session ou une demande de déchiffrement, le système d'authentification ou de vérification accède automatiquement à une liste de révocation à jour depuis une autorité de certification approuvée afin de vérifier la validité du certificat.

Si ce dernier apparaît dans la liste, l'accès est purement et simplement refusé. Si la CRL renseignée dans le certificat n'est pas joignable, l'accès est également refusé.

- **Répondeurs en ligne** : les clients peuvent utiliser un répondeur en ligne afin d'interroger une autorité de certification directement pour connaître l'état d'un certificat venant d'être présenté.

Ce processus est plus rapide que le téléchargement et la consultation d'une nouvelle liste de révocation des certificats.

Depuis les précédentes versions de Windows Server, Microsoft intègre dans son système d'exploitation un composant permettant l'installation, la configuration et la gestion d'autorités de certification. Depuis Windows Server 2008, ce composant apparaît comme un rôle de serveur, plus connu sous le nom d'Active Directory Certificate Services (AD CS).

III.3. Présentation d'AD CS

Les services de certificats Active Directory dans Windows Server 2012 R2 sont représentés sous la forme d'un rôle de serveur. Ce rôle permet notamment de mettre en place une infrastructure hiérarchique à clés publiques afin de créer et gérer des certificats. Grâce à AD CS, il est possible de répondre à différents besoins de l'entreprise tels que sécuriser les communications VPN, WAN, LAN et sans fil, sécuriser des sites web IIS ou même sécuriser les messages électroniques issus d'un serveur de messagerie Microsoft Exchange.

III.3.1. Services de certificats AD CS

Il est possible d'installer AD CS comme étant une autorité de certification autonome ou d'entreprise. AD CS peut fournir des services de certificats à l'intérieur et à l'extérieur du réseau de l'entreprise. AD CS intègre des fonctionnalités supplémentaires avec le système d'exploitation Windows Server 2012.

III.3.1.1. CA autonome

Une autorité de certification autonome a l'avantage de pouvoir être installée sur un serveur membre d'un domaine ou d'un groupe de travail. Le rôle AD DS n'est donc pas indispensable lorsque l'on souhaite déployer une CA autonome. Ce type d'installation d'autorité de certification est généralement utilisé dans des infrastructures à clés publiques à plusieurs couches. L'autorité de certification racine installée dans une hiérarchie de CA doit être de type autonome. Sa fonction est de générer un certificat à destination des CA intermédiaires. Après l'installation d'une CA racine autonome, il est conseillé de la mettre hors ligne ou hors réseau afin d'accroître la sécurité de l'infrastructure à clé publique. Dans une infrastructure de CA à trois couches, l'autorité de certification intermédiaire doit également être de type autonome.

L'installation d'une autorité de certification racine nécessite la configuration des éléments suivants :

- **Points de distribution de liste de révocation des certificats** (CDP : *CRL Distribution Points*) : les CDP permettent d'indiquer l'emplacement d'une CRL, utilisé lors d'une demande de validation d'un certificat.
- **Accès aux informations de l'autorité de certification** (AIA : *Authority Information Access*) : les AIA permettent d'indiquer aux utilisateurs l'emplacement où trouver le certificat de l'autorité de certification racine.

III.3.1.2. CA d'entreprise

Une autorité de certification d'entreprise est intégrée aux services de domaine Active Directory. Dans une infrastructure à clés publiques, les CA d'entreprise sont généralement utilisées en tant qu'autorité de certification émettrice. L'installation de ce type de CA est donc réalisée sur un serveur membre d'un domaine. Ainsi, les utilisateurs du domaine peuvent soumettre des demandes de certificats qui peuvent être approuvées directement par l'autorité de certification d'entreprise en charge de délivrer des certificats.
L'installation de ce type de CA est dépendante du système d'exploitation qui les héberge.

III.3.2. Gestion d'AD CS

L'installation du rôle de serveur AD CS ajoute dans les outils d'administration du système d'exploitation la console d'administration **Autorité de certification** :

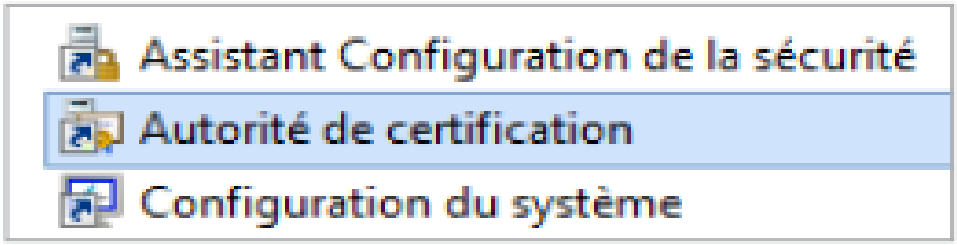


Figure III.2. Console d'administration Autorité de certification.

Cette console de gestion est un composant logiciel enfichable situé dans :
%SYSTEMROOT%\system32\certsrv.msc
L'installation d'AD CS installe également le service Windows **Services de certificats Active Directory**, configuré avec un type de démarrage **Automatique** :

Nom	Description	État	Type de démarrage
Services de chiffrement	Fournit trois...	En cours d'ex...	Automatique
Services de certificats Active Directory	Crée, gère e...	En cours d'ex...	Automatique
Services Bureau à distance	Autorise les ...		Manuel

Figure III.3. Console de configuration automatique pour le démarrage d'ADCS.

Si vous venez d'installer une autorité de certification intermédiaire, le service Windows **Services de certificats Active Directory** sera dans l'état arrêté car aucun certificat issu d'une autorité de certificatparente n'aura été importé.

La console de gestion **certsrv**permet de visualiser l'état du serveur à l'aide d'une icône sur l'autorité de certification (*Service Windows arrêté ou démarré*). Il est également possible d'arrêter ou démarrer une autorité de certification depuis le composant logiciel enfichable :

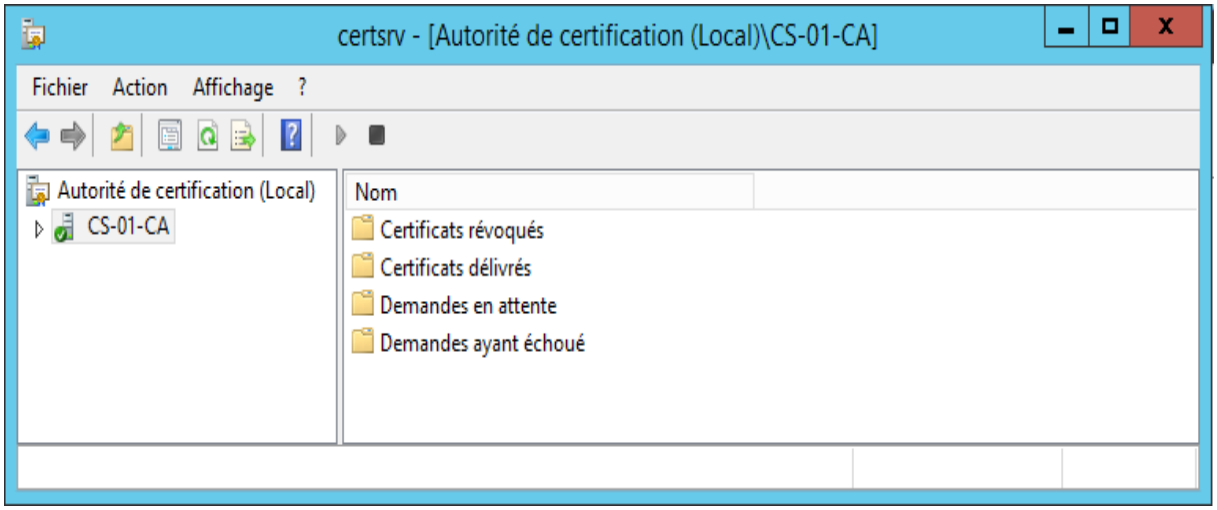


Figure III.4. Console de démarrage ou l'arrêt d'une CA.

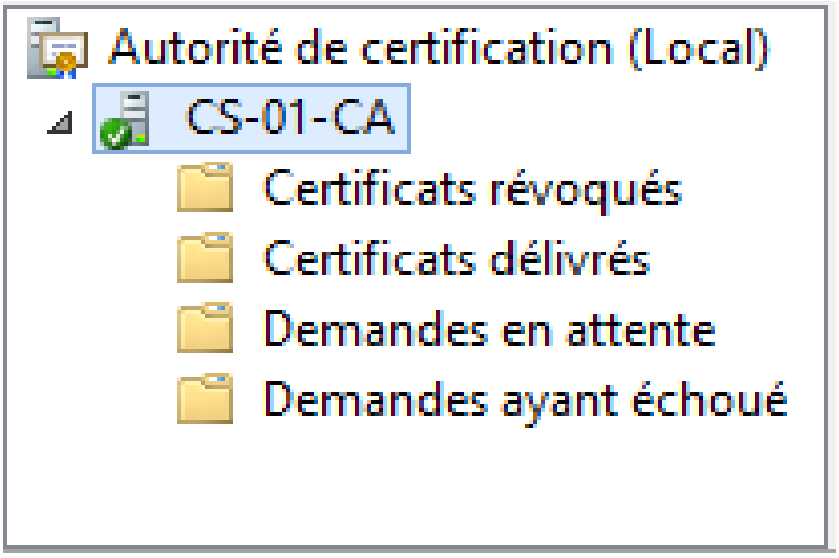


Figure III.5. L'arborescence de la console de gestion AD CS.

L'arborescence de la console de gestion AD CS offre l'accès aux conteneurs suivants :

- **Certificats révoqués** : permet de visualiser l'ensemble des certificats révoqués ou expirés de votre organisation.
- **Certificats délivrés** : permet de visualiser l'ensemble des certificats approuvés et délivrés par votre organisation.
- **Demandes en attente** : permet de visualiser l'ensemble des certificats demandés par des composants du réseau ou autres utilisateurs. Les certificats figurant dans ce conteneur doivent être approuvés manuellement avant de pouvoir être utilisés.
- **Demandes ayant échoué** : permet de visualiser l'ensemble des demandes de certificats ayant échoué. À partir de ce conteneur, il est également possible d'émettre des certificats.

Lorsqu'un certificat est installé, ce dernier peut être géré ou visualisé depuis la console de gestion des certificats :

- **%SYSTEMROOT%\system32\certlm.msc**: permet de gérer les certificats liés à l'ordinateur.
- **%SYSTEMROOT%\system32\certmgr.msc**: permet de gérer les certificats liés à l'utilisateur.

III.3.3. Hiérarchie de CA

Une infrastructure à clés publiques peut fonctionner à l'aide d'un seul serveur autonome hébergeant le rôle de serveur AD CS. Cependant, lorsque plusieurs autorités de certification composent votre infrastructure, on parle alors de hiérarchie de CA. L'utilisation d'un serveur unique pour l'exploitation des services de certificats peut affaiblir la sécurité de l'infrastructure à clés publiques. Afin de renforcer la sécurité, il est possible d'ajouter des CA supplémentaires à une PKI. Il existe différents types de hiérarchie de CA dont la topologie correspondra à la sécurité ou aux besoins de l'entreprise.

Dans une hiérarchie de CA à plusieurs couches, un serveur CA racine est chargé d'attribuer des certificats à des CA secondaires, afin qu'elles puissent elles-mêmes attribuer des certificats aux utilisateurs, ordinateurs ou services. Dans une hiérarchie de CA, le plus haut niveau est nommé autorité de certification racine. Toute attaque contre le serveur hébergeant la CA racine, ou une CA quelconque, peut compromettre la sécurité de l'ensemble des CA intermédiaires ou sous-jacentes. C'est pourquoi il est important de sécuriser les autorités de certification de plus haut niveau en les mettant hors ligne ou inaccessibles à toutes tentatives d'intrusions. Dans la plupart des situations, une PKI est implémentée dans une infrastructure à deux ou trois couches.

III.3.3.1. Infrastructure à deux couches

Une infrastructure d'autorité de certification à deux couches apporte un minimum de sécurité car la CA racine peut être placée hors ligne afin de ne conserver que la CA émettrice pour l'attribution et l'approbation des certificats. Généralement, une CA racine autonome est mise en place au plus haut niveau et une CA d'entreprise émettrice est située au plus bas niveau pour fournir des services de certificats aux machines ou utilisateurs du réseau.

Selon le modèle d'infrastructure à deux couches, les types d'autorités de certification suivantes sont installés :

- **CA racine** : CA autonome (hors ligne)
- **CA émettrice** : CA d'entreprise (en ligne)

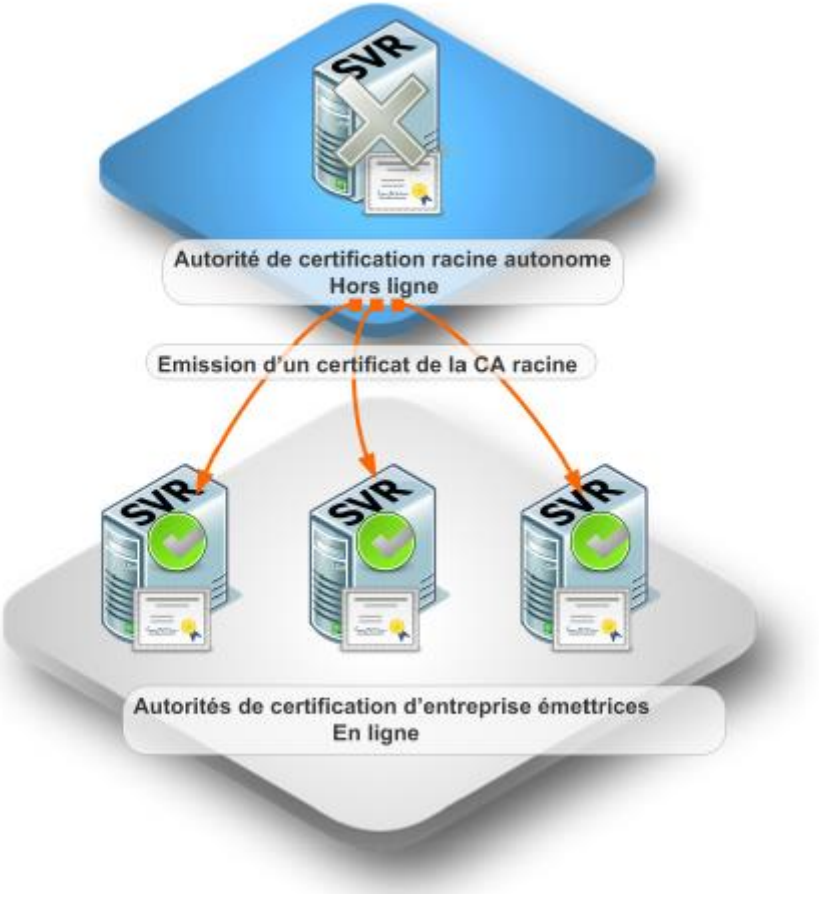


Figure III.6. Infrastructures d'une CA à deux couches.

III.3.3.1.1. CA racine

La CA racine est le plus haut niveau d'une hiérarchie de CA. Lors de la création d'une nouvelle autorité de certification racine, il faut impérativement créer une nouvelle clé privée. La création d'une clé privée nécessite de renseigner les informations suivantes :

- **Sélection d'un fournisseur de chiffrement** : pour générer une paire de clés pour l'autorité de certification racine, l'API Microsoft Crypto doit utiliser un fournisseur de chiffrement logiciel ou matériel.
- **Indication d'une longueur de clé** : le nombre de caractères de la clé détermine la longueur des clés de la paire. Plus la longueur de clé est élevée et plus le traitement pour décoder est long.
- **Sélection d'un algorithme de hachage** : les algorithmes de hachage permettent de sécuriser les clés en utilisant un algorithme de calcul spécifique, destiné à générer des informations chiffrées aux paires de clés.

Dans notre application nous avons utilisé :

RSA#Microsoft Software Key Storage Provider come fournisseur de chiffrement, une longueur de clés de 4096 bits et un algorithme de hachage: SHA1

III.3.3.1.2. CA émettrice

Une CA émettrice doit être une CA d'entreprise, installée dans le but de délivrer des certificats aux utilisateurs, serveurs ou autres composants ayant fait la demande. L'installation d'une CA d'entreprise permet l'auto inscription des utilisateurs ou ordinateurs.

III.3.4. Services de rôles AD CS

Le rôle de serveur AD CS possède six services de rôle sous Windows Server 2012 R2, contre quatre sur Windows Server 2008 R2.

III.3.4.1. Autorité de certification

Le service de rôle **Autorité de certification** a pour objectif de réaliser les missions suivantes :

- Accepter les demandes de certificats.
- Délivrer des certificats.
- Révoquer des certificats.
- Publier des listes de révocation de certificats.

L'installation de ce rôle de service AD CS nécessite des privilèges d'administration locale lorsque le serveur est membre d'un groupe de travail, ou des privilèges d'administration de domaine si le serveur est membre d'un domaine Active Directory. Pour installer ce service de rôle en tant qu'autorité de certification d'entreprise, le serveur doit être membre d'un domaine Active Directory.

III.3.5. Certificats

Un certificat numérique peut être délivré par une autorité de certification interne ou publique.

Cependant, si aucune autorité de certification n'est disponible, il est également possible de générer un certificat auto-signé. Par défaut, tout système Microsoft intègre des outils permettant de générer des certificats auto-signés. Un certificat auto-signé signifie que le système s'attribue lui-même un certificat qu'il reconnaît étant valide par la même occasion. Un certificat auto-signé n'est reconnu que par le système l'ayant généré.

Par exemple, prenons le cas d'un administrateur mettant en place un site web IIS dont l'accès doit se faire via le protocole HTTPS en utilisant un certificat auto-signé. Les utilisateurs accédant au site web obtiennent systématiquement un message d'avertissement de leur navigateur, indiquant que le certificat présenté par le site n'est pas vérifié et que l'accès est potentiellement dangereux. Afin que les utilisateurs n'obtiennent plus ce message, il faut que chacun d'entre eux installe le certificat auto-signé par IIS, dans leur magasin local **Autorité de certification racine de confiance**.

La création des certificats en environnement Microsoft est majoritairement basée sur le standard X.509.

Un certificat contient plusieurs données telles que :

- Des informations d'identifications (Nom, raison sociale, localisation, etc.).
- Une clé publique.
- L'algorithme de hachage/Empreinte numérique.
- L'identité de l'émetteur.
- Les dates de validité.
- Un numéro de série.
- La version du certificat.
- L'URL d'accès aux informations de l'autorité de certification.
- Les URL des points de distribution de listes de certificats révoqués.

III.3.5.1. Modèles de certificats

Les modèles de certificats servent à générer des certificats à partir de modèles prédéfinis qu'il est possible de personnaliser au préalable. Ces modèles ne sont disponibles que sur un serveur hébergeant le rôle AD CS en tant qu'autorité de certification d'entreprise émettrice.

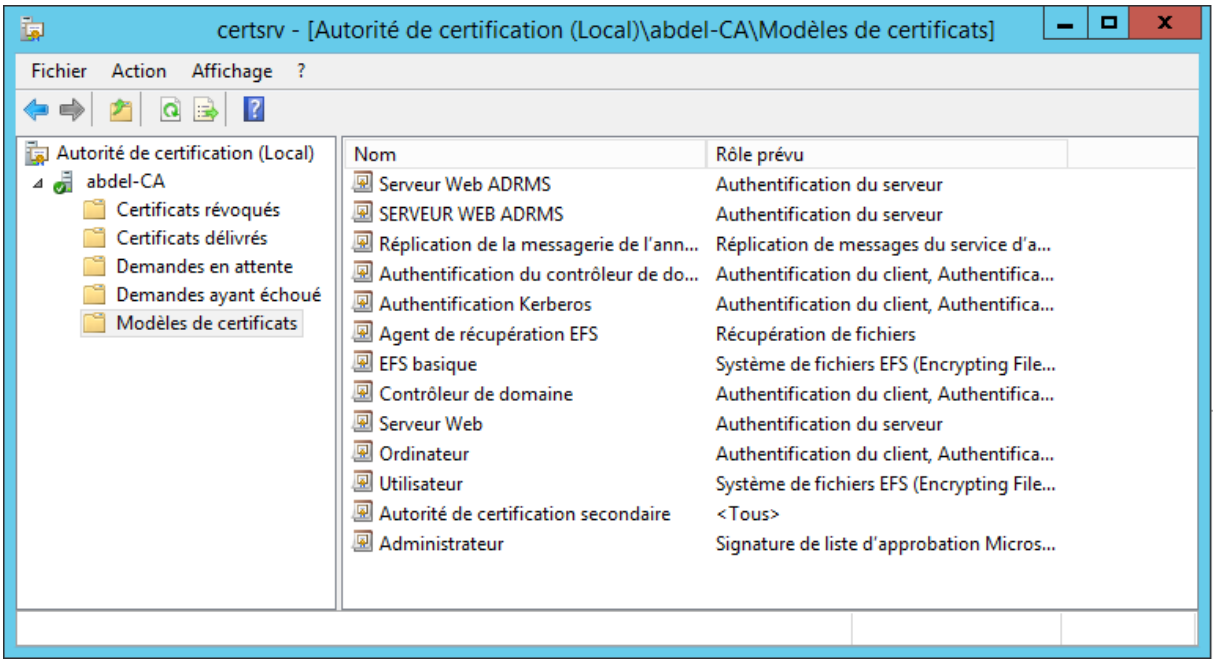


Figure III.7. Liste de modèles de certificats.

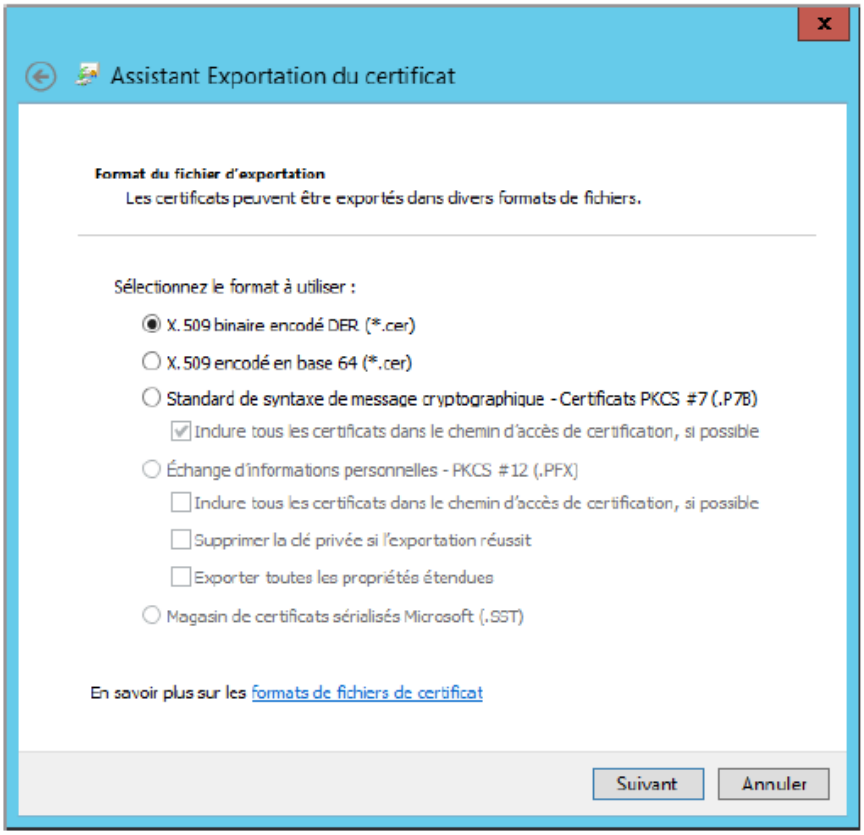
Il est également possible de créer ou dupliquer des modèles de certificats afin de compléter la liste existante.

Pour qu'un utilisateur puisse avoir le droit de sélectionner un modèle de certificat, il faut lui attribuer des droits d'inscription pour le certificat spécifique.

Pour gérer les modèles de certificats, il suffit de faire un clic droit sur le conteneur **Modèles de certificats** dans la console **Autorité de certification**, puis de cliquer sur **Gérer**.

Lorsqu'une demande de certificat a été validée et délivrée, il est possible d'extraire le certificat de l'autorité de certification émettrice afin de l'exploiter. Pour cela, il suffit d'utiliser l'Assistant Exportation du certificat et sélectionner un des formats que l'on souhaite utiliser pour générer le certificat.

Il existe différents types de fichiers de certificat :



*

Figure III.8. Format de fichier du certificat.

- **X.509 binaire encodé DER (*.cer)** : les fichiers X.509 utilisent une norme de cryptographie définie par l'Union internationale des communications. Le format de fichier DER (*DistinguishedEncodingRules*) est souvent utilisé pour les postes exécutant un système d'exploitation autre que Windows.

- **X.509 encodé en base 64 (*.cer)** : ce type de format est souvent utilisé dans les infrastructures de messagerie car la norme de cryptographie S/MIME (*Secure/Multipurpose Internet Mail Extension*) est supportée.

- **Standard de syntaxe de message cryptographique - Certificats PKCS #7 (*.p7b)** : ce format de fichier est souvent utilisé pour le transfert d'un certificat d'un ordinateur à un autre.

- **Échange d'informations personnelles - PKCS #12 (*.pfx)** : ce format de fichier est également utilisé pour le transfert d'un certificat d'un ordinateur à un autre mais en incluant la clé privée et la clé publique, ce qui peut être dangereux pour l'intégrité des données à protéger.

- **Magasin de certificats sérialisés Microsoft (*.SST)** : ce format de fichier est souvent utilisé pour le transfert d'un certificat racine vers un autre ordinateur.

III.3.5.2. Demande de certificat

Il existe plusieurs méthodes pour demander un certificat à une autorité de certification. Il est possible d'effectuer une demande de certificat depuis :

- L'interface web de l'autorité de certification, en accédant au répertoire virtuel Certsrv.
- Le composant logiciel enfichable Certificats.

Lors d'une demande de certificat, si l'utilisateur est habilité, il peut également sélectionner un modèle de certificat préconfiguré pour un usage bien défini (exemple : Certificat utilisateur, ordinateur, Basique EFS, etc.).

Si l'auto-inscription n'est pas activée, l'utilisateur devra soumettre systématiquement une demande de certificat qui sera validée manuellement par l'administrateur de l'autorité de certification. Si l'auto inscription est activée, la demande de certificat sera acceptée et délivrée automatiquement.

III.4. Discussion

Les infrastructures à clés publiques (**PKI**) ont été créées dans le but de confirmer l'identité de chaque composant d'un réseau informatique lors d'un échange de données. Elles nous aident donc à sécuriser les communications informatiques ou connexion, effectuer un paiement en ligne sur un site commercial et notamment à établir des liens de confiances entre des personnes physiques et des certificats numériques tels que le ferait une carte nationale d'identité.

Ce chapitre donne un petit aperçu sur les infrastructures à clé publique et leurs composants. Il traite aussi des certificats et leurs modèles ainsi que les différentes méthodes de demande de certificat.

IV.1. Préambule

Dans l'aire ou nous vivons, les périphériques mobile sont devenues très présents dans la vie quotidienne en tant que professionnel et personnel et cela engendre différents risques de sécurité.

Pour cela, nous devons mettre en œuvre un environnement sécurisé en utilisant différents moyens et outils pour protéger et garantir les données de l'entreprise, tout en facilitant le travail aux employés (utilisateurs) en utilisant des certificats.

Dans ce chapitre nous avons expliqué les différentes méthodes et matériels utilisés pour la mise en œuvre de l'architecture système de notre entreprise.

IV.2. Description de projet

L'objectif du projet est d'assurer un environnement de travail aux utilisateurs en leur offrant un système, d'accès à des ressources en toute sécurité via des clés.

IV.3. Objectif de l'application

Chaque application ou conception a un but et objectif bien précis, la nôtre c'est d'offrir aux clients un environnement qui va leurs permettre de travailler en toute sécurité et simplicité, en leur imposant une certaine politique associée aux outils utilisés tel que :

- ✓ Active Derictory : Création de session des utilisateurs.
- ✓ Clé et certificats.
- ✓ GPO : Politique des restrictions pour les utilisateurs.
- ✓ DHCP : Fournir la configuration IP.
- ✓ DNS : consiste à trouver l'adresse IP d'un ordinateur à partir de son nom.

IV.4. Outils

- ✓ Serveur 2012 R2.
- ✓ Windows8.
- ✓ VM Worckstation pro V.14.

IV.4.1. Environnement matériel

- ✓ Micro-ordinateur.

IV.4.2. Environnement de développement

Nous avons choisi un environnement de Windows pour réaliser notre projet, ce dernier, assure la gestion de l'ordinateur et des périphériques. Les systèmes d'exploitation interactifs offrent à l'utilisateur un environnement de travail intuitif, performant et la possibilité d'assurer la sécurité à nos utilisateurs et leurs fournir les services souhaités.

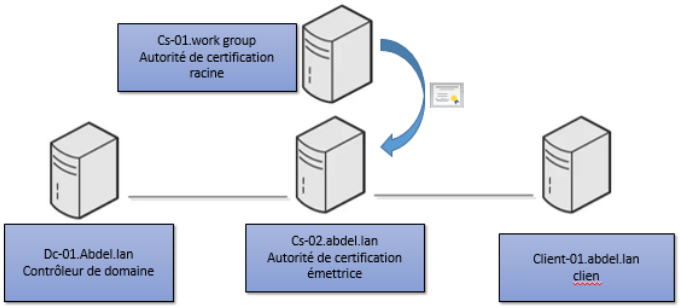


Figure IV.1. Architecture d'autorité de certification

IV.5. Autorité de certification Autonome

IV.5.1. Installation de l'autorité de Certification Autonome

- **Étape 1** : ouvrir une session sur le serveur **CS-01** avec des identifiants d'administration, puis dans le **Gestionnaire de serveur**, nous avons clique sur **Ajouter des rôles et des fonctionnalités**.
- **Étape 2** : Nous avons cliqué sur **Suivant** pour passer les pages **Avant de commencer**, **Sélectionné le type d'installation** et **Sélectionné le serveur de destination**.
- **Étape 3** : dans l'étape **Sélectionner des rôles de serveurs**, nous avons coche la case correspondant au rôle **Services de certificats Active Directory**, qui vont servir à créer les autorités de certification ainsi que les rôles associer afin d'émettre et gérer les certificats utilisées dans plusieurs applications. Ensuite clic sur **Suivant**

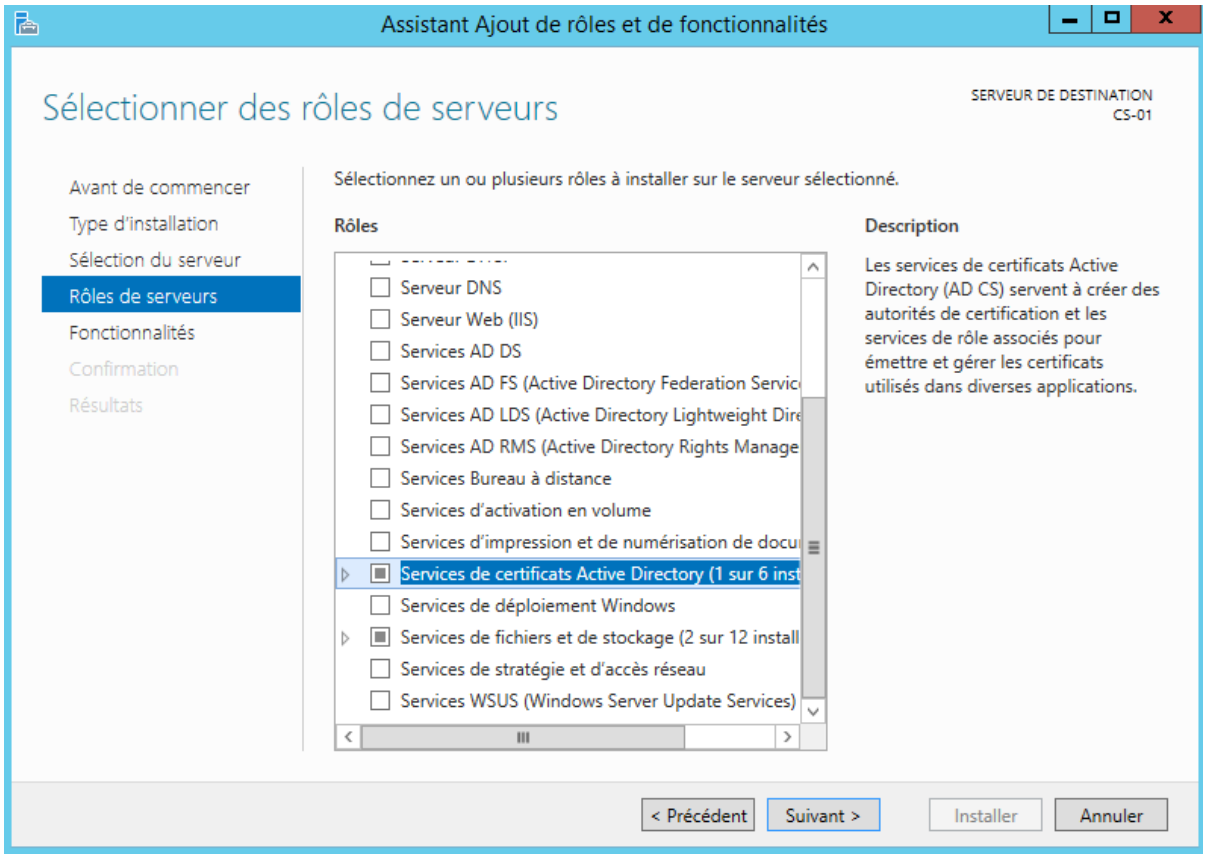


Figure IV.2. Service de certificat Active Directory.

- **Étape 4** : dans l'étape **Services de rôle**, nous avons coché la case **Autorité de certification**, puis **Suivant** :

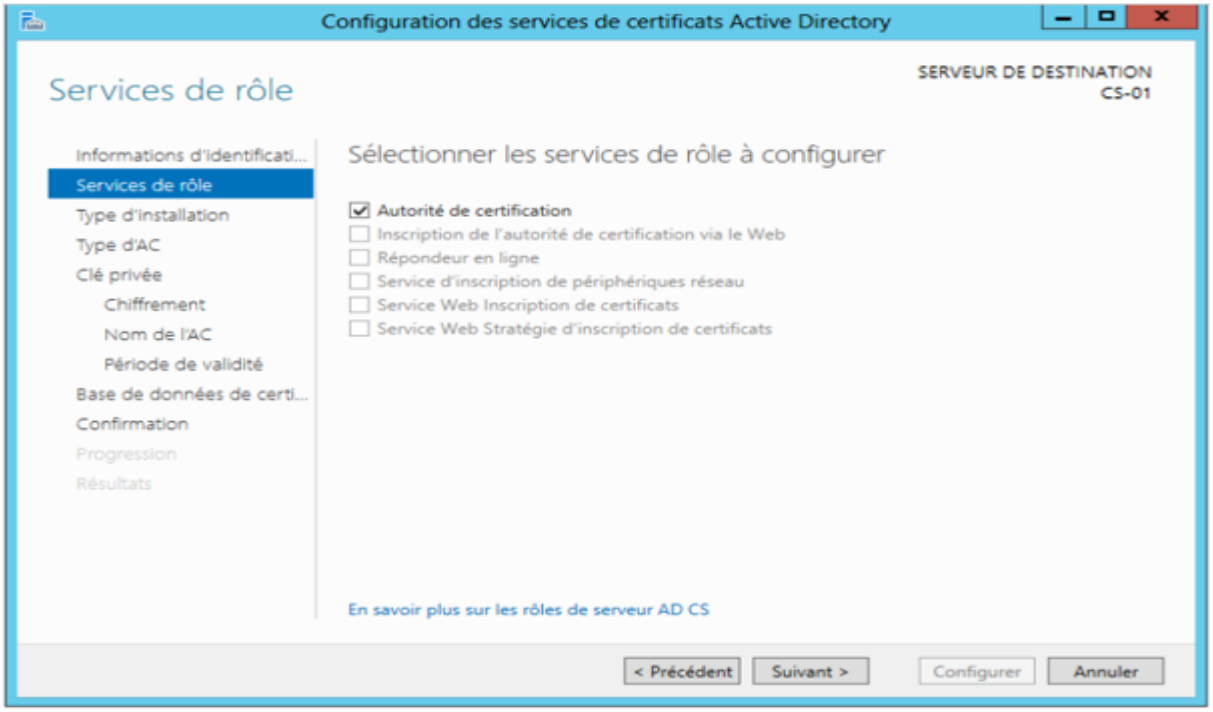


Figure IV.3. Autorité de certification.

- **Étape 5** : dans l'étape **Type d'installation**, nous avons coché sur Autorité de certification autonome, sachant que les certificats autonomes ne nécessitent pas un ADDS elles peuvent être membre d'un groupe de travail ou d'un domaine et peuvent être utilisé sans connexion (hors connexion). Suivant.

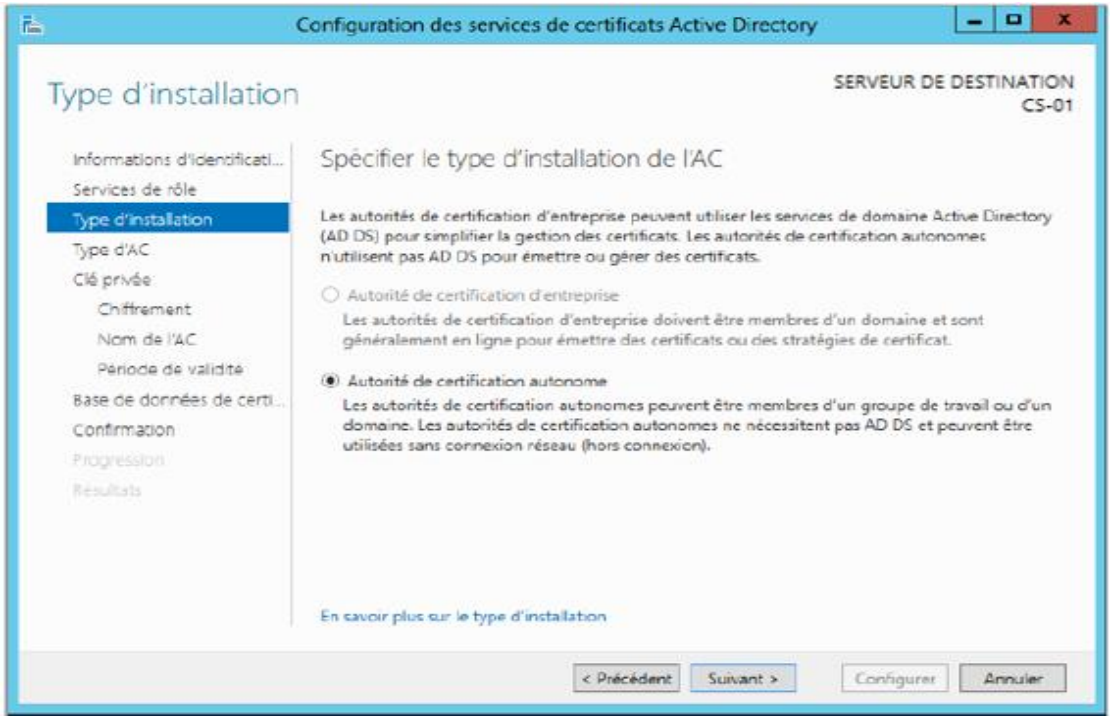


Figure IV.4. Spécifier le type de CA à installer.

- **Étape 6** : lors de l'installation des services AD CS, une hiérarchie à clé publique (PKI) est créée. Dans l'étape spécifier le type d'AC nous avons coché sur **Autorité de certification racine** qui émet ses propres certificats auto-signés, structuré au sommet de la hiérarchie PKI.

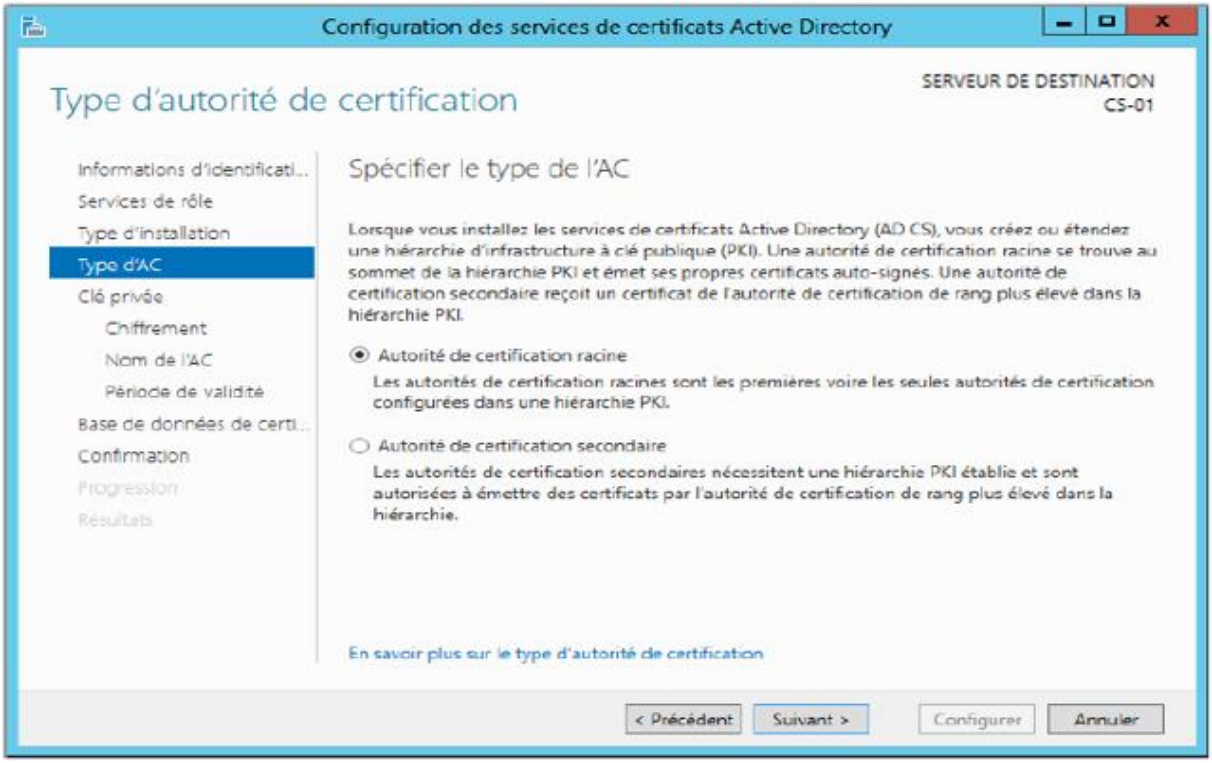


Figure IV.5. Choix du type de CA.

- **Étape 7** : dans l'étape de Spécifier le type de la clé privée, nous avons coché sur **créer une clé privée** afin de créer une nouvelle.

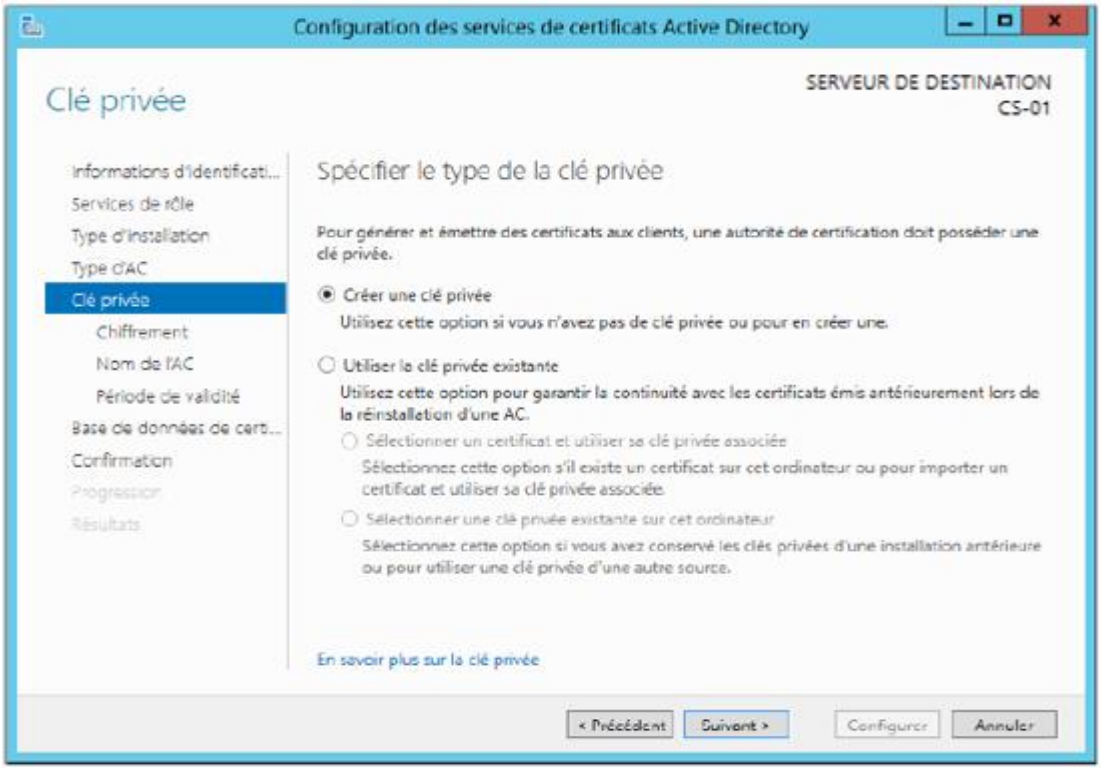


Figure IV.6. Choix du type de clé privée.

- **Étape 8** : dans l'étape **Chiffrement pour l'autorité de certification**, nous avons sélectionné le fournisseur de chiffrement **RSA#Microsoft Software Key Storage Provider** dont la longueur de clé de *2048* et l'algorithme de hachage **SHA1.**

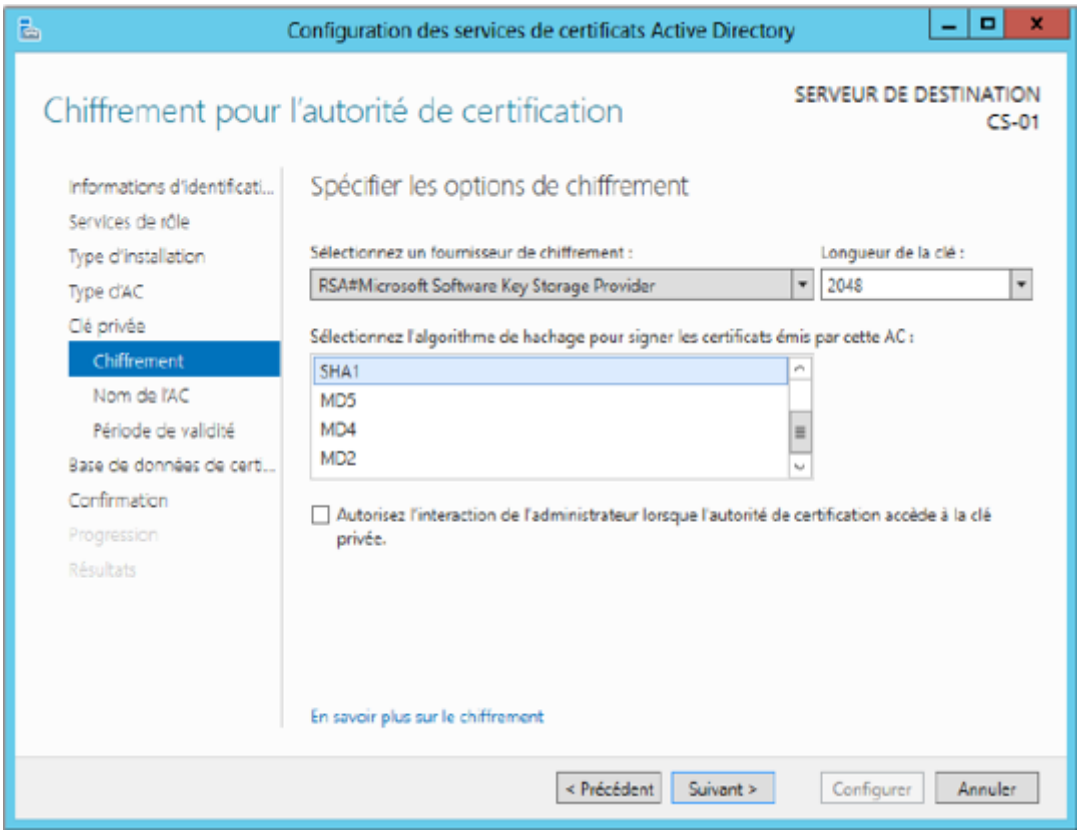


Figure IV.7. Choix du chiffrement.

- **Étape 9** : dans cette étape nous allons **s**pécifier le **Nom de l'autorité de certification**, en écrivant le nom commun *abdel-Root-CA* dans le champ**Nom commun de cette AC**, afin d'identifier cette CA ce nom est ajouté à tous les certificats émis par l'AC.

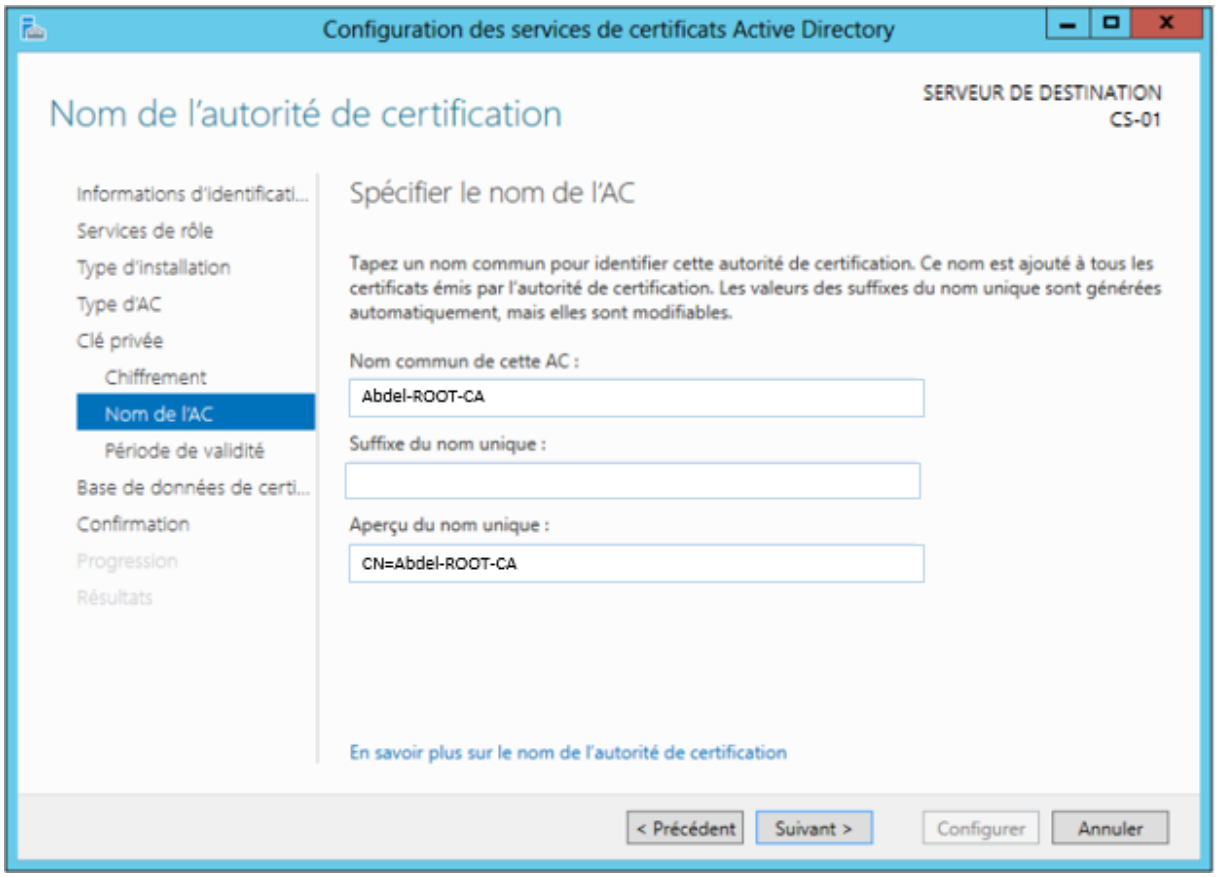


Figure IV.8. Spécifier le nom de l'AC.

- **Étape 10** : dans l'étape **Base de données de l'autorité de certification**, nous avons spécifié l'emplacement de la base de données de certificats dans le répertoire *D:\CA\CertDB*, et l'emplacement du journal de la base de données de certificats à l'emplacement : *D:\CA\CertLog*.

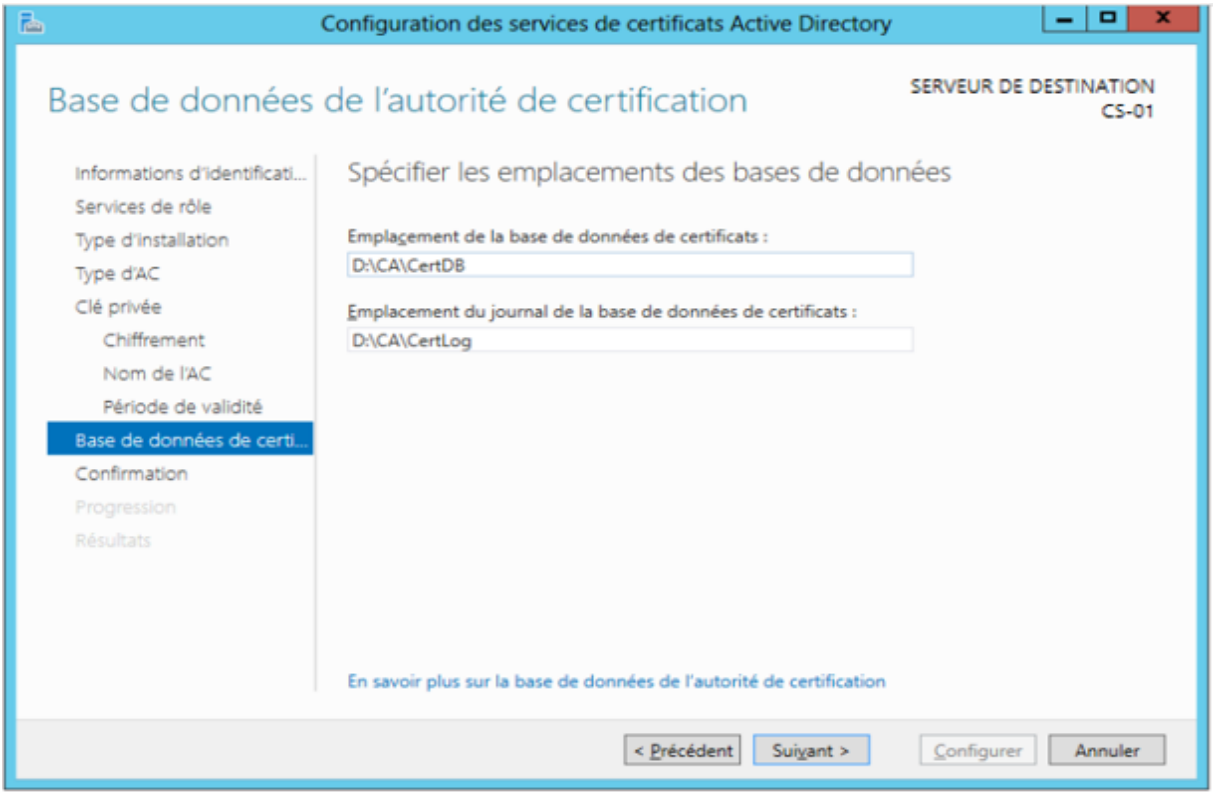


Figure IV.9. Emplacement de la Base de données de l'autorité de certification, et journal de la base de données de certificats

- Dans l'étape de **Confirmation**, nous devons vérifier que les informations de configuration de votre autorité de certification racine sont justes, puis nous cliquons sur **Configurer**.
- Dans l'étape **Résultats**, nous devons nous assurer que notre autorité de certification affiche un état **Configuration réussie**. Pour pouvoir fermer.

IV.5.2. Configuration de l'autorité de Certification Autonome

- Dans les outils d'administration nous avons démarré la console **Autorité de certification**, Dans l'arborescence de la console, nous avons fait un clic droit sur l'autorité de certification **abdel-Root-CA** puis un clic sur **Propriétés**.
- Nous avons clique sur l'onglet **Extensions**, puis nous avons cliqué sur **Ajouter** a fin d'ajouter un nouveau point de distribution de liste de révocation des certificats.
- Dans le champ **Emplacement**, nous avons tapé l'URL se trouvant en dessous ensuite cliqué sur **OK** :

http://CS-02.abdel.lan/CertData/<NomAutoritéCertification><SuffixeNomListeRévocationCertificats><ListeRévocationCertificatsDeltaAutorisée>.crl

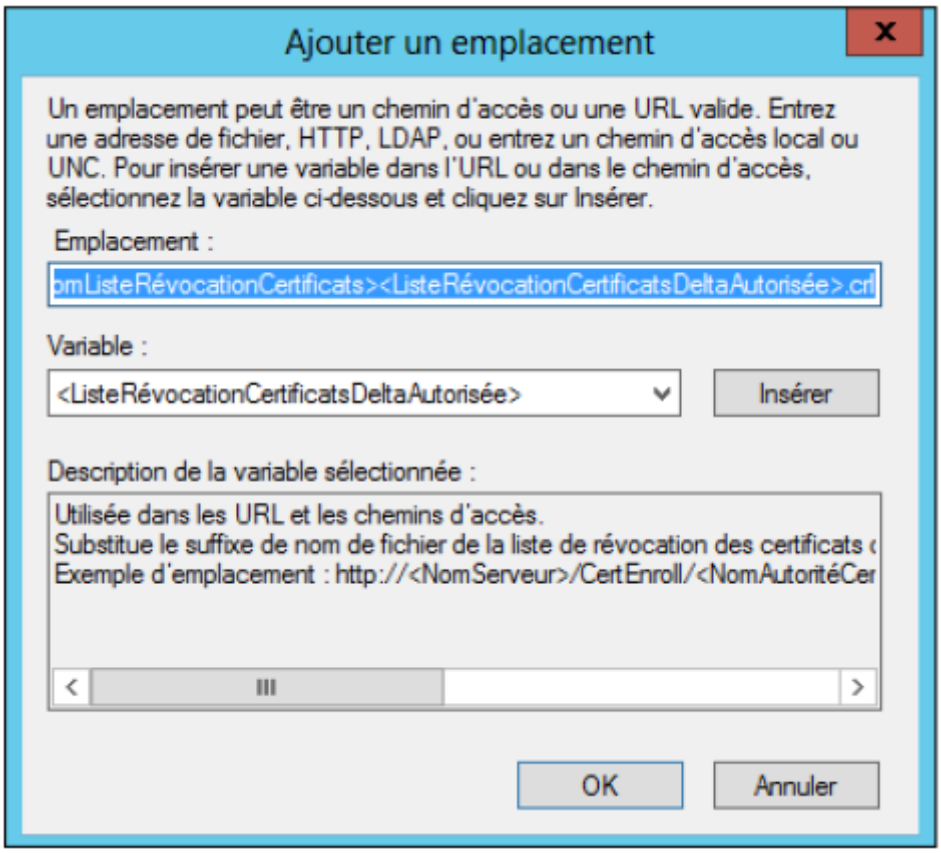


Figure IV.10. Ajout de l'emplacement.

- **Étape 11** : Dans cette étape nous avons coché les cases suivantes :

- Inclure dans les listes de révocation des certificats afin de pouvoir rechercher les listes de révocation des certificats delta

- Inclure dans l'extension CDP des certificats émis

Puis un clic sur **Appliquer**.

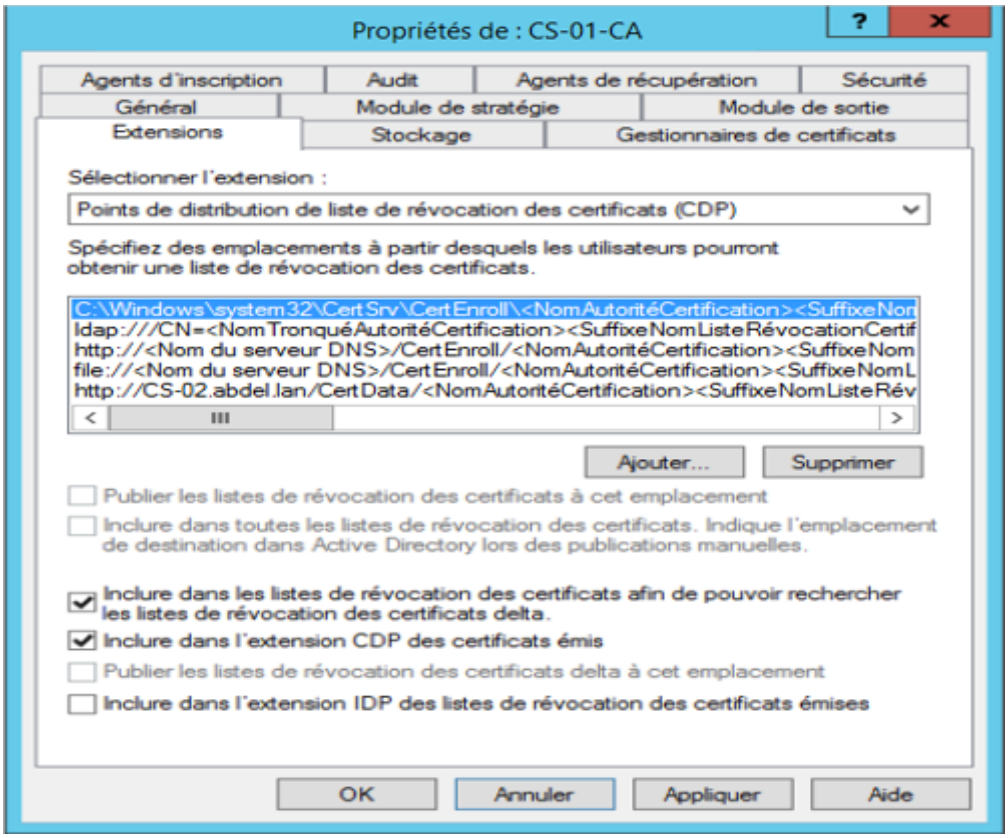


Figure IV.11. Propriétés de la CA.

- **Étape 12** : Dans le champ **Emplacement**, nous avons tapé l'URL suivante : *http://CS-02.abdel.lan/CertData/<Nom du serveur DNS><NomAutoritéCertification><NomCertificat>.crt*

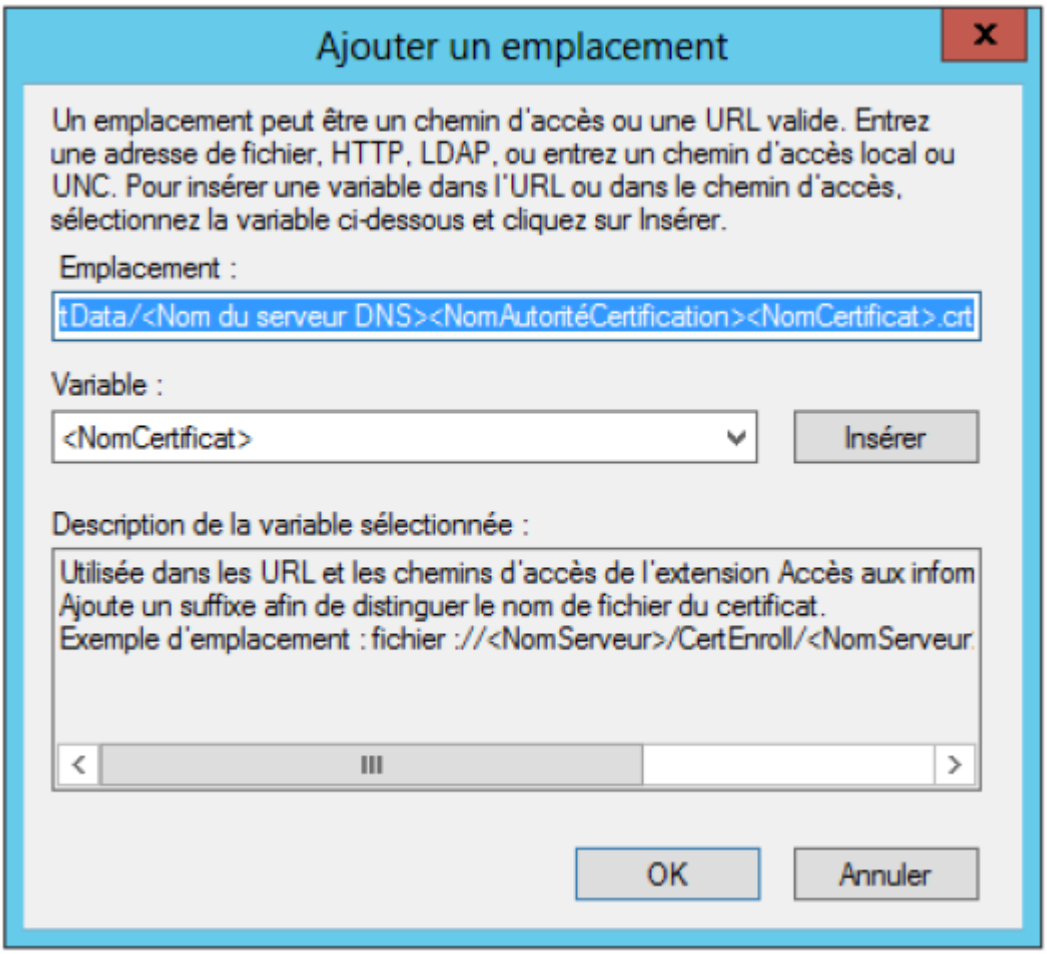


Figure IV.12. Ajout d'un emplacement.

- **Étape 13** : Dans cette étape nous avons coché la case **Inclure dans l'extension AIA des certificats émis** puis sur **OK**.

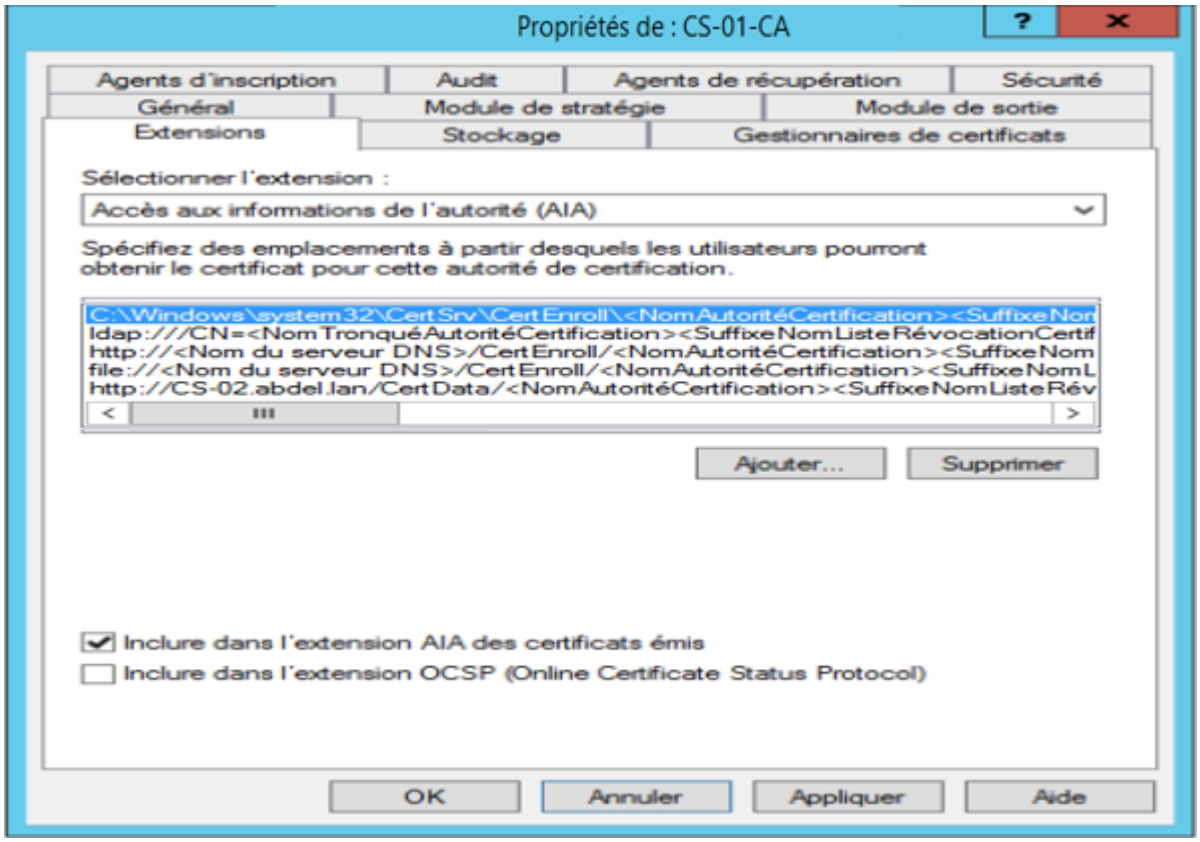


Figure IV.13. Choix de l'extension.

- **Étape 14** : Dans cette étape nous allons publier les certificats révoqués pour se fait, sur la console de gestion certsrv, nous avons développé l'arborescence, sélectionné le dossier **Certificats révoqués**, avec un clic droit dessus, puis clic sur **Toutes les tâches** à la fin nous avons cliqué sur **Publier** :

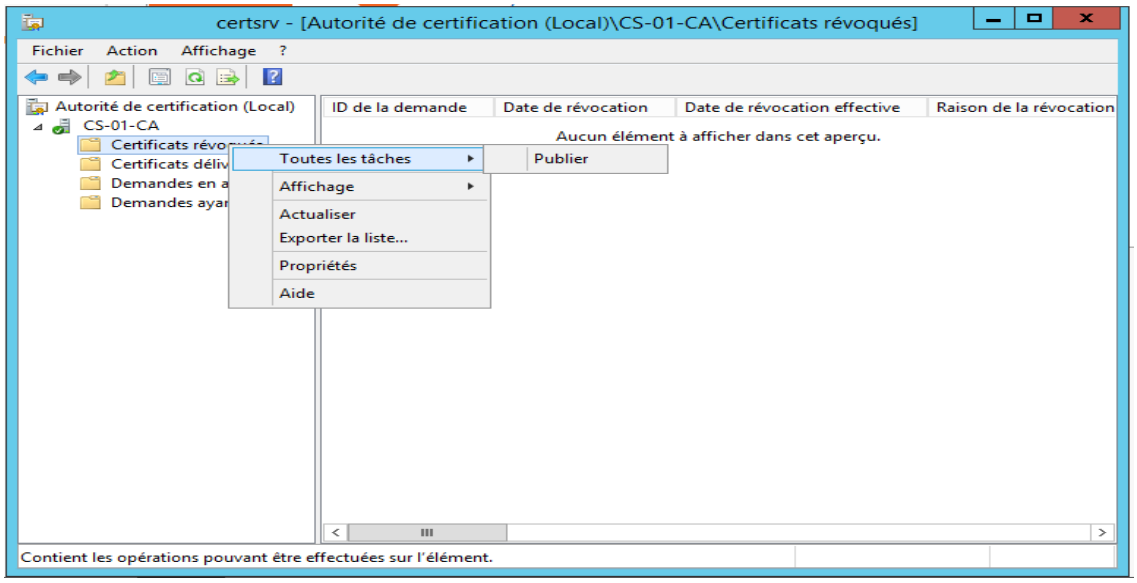


Figure IV.14. Publier les certificats révoqués

Étape 15 : Dans cette étape nous avons copié les deux fichiers présents dans le répertoire C:\Windows\System32\Certsrv\CertEnroll, sur le répertoire *C:\Partage* se trouvant sur le serveur CS-02.

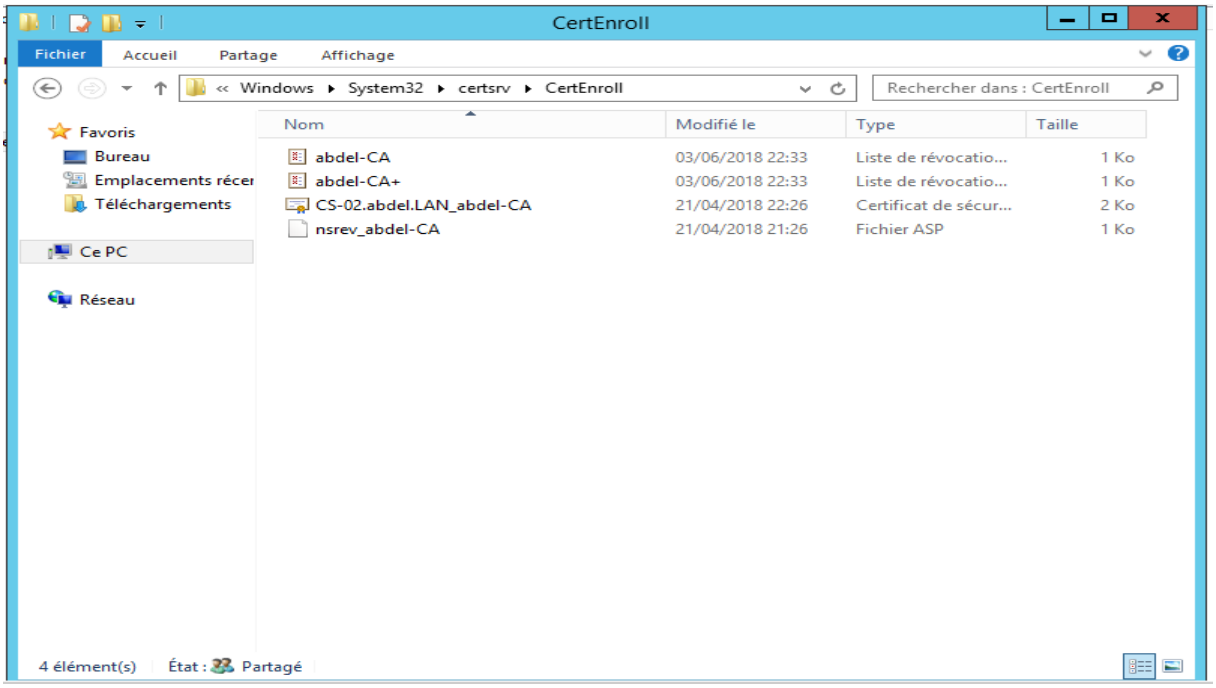


Figure IV.15. Copier les fichiers.

- **Étape 16** : Dans cette étape nous allons modifier les paramètres de partage pour cela, nous sommes allés dans le panneau de configuration du serveur **CS-02,** nous avons cliqué sur **Afficher l'état et la gestion du réseau**, puis clic sur **Modifier les paramètres de partage avancés**.
- **Étape 17** : dans la section **Invité ou public**, nous avons sélectionné **Activer le partage de fichiers et d'imprimantes** ensuite cliqué sur **Enregistrer les modifications** :

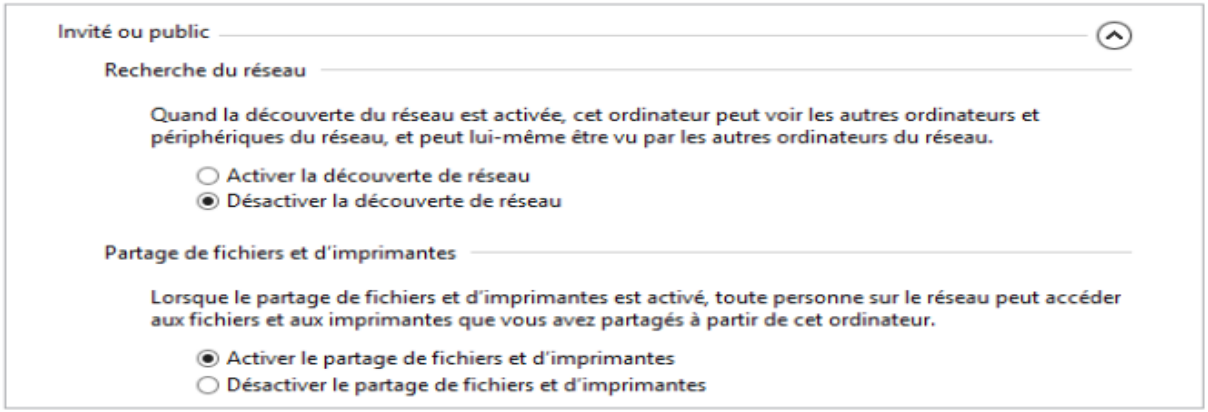


Figure IV.16. Partages de fichiers et imprimantes.

- **Étape 18** : nous avons ouvert une session sur le serveur DC-01 avec des privilèges d'administration dans le domaine abdel.lan. Nous avons démarré la console de gestion DNS puis nous avons créé l'hôte A dans la zone de recherche directe d'abdel.lan :

Nom : CS-01

Adresse IP : 192.168.0.112

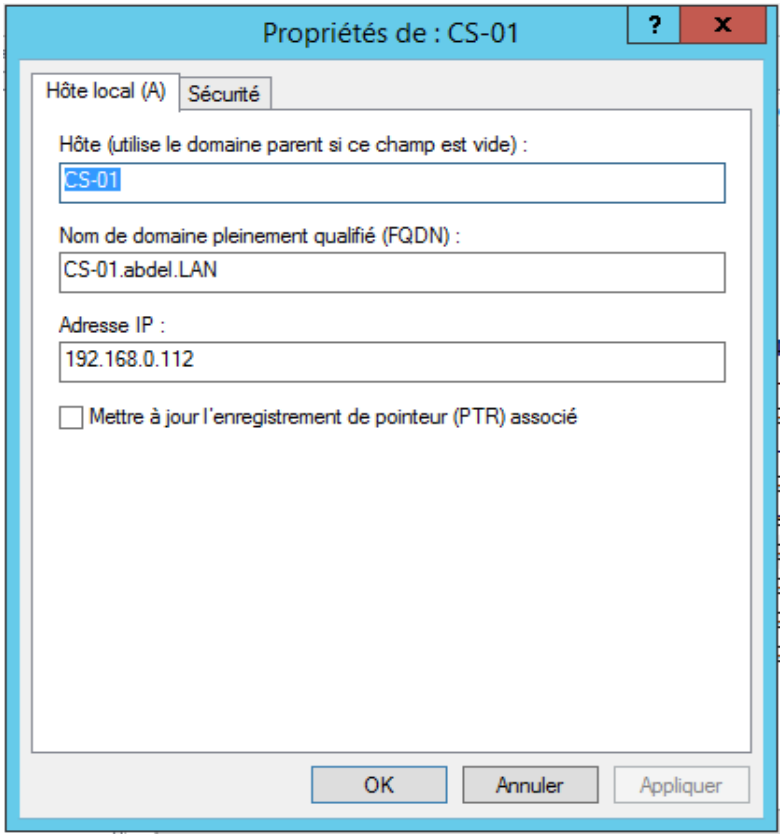


Figure IV.17. Création de l'hôte A.

- ✓ *À ce stade, notre autorité de certification racine est désormais installée et configurée.*

IV.6. Autorité de certification d'entreprise (émettrice)

IV.6.1. Installation de l'autorité de Certification d'entreprise

- **Étape 1** : nous avons ouvert une session sur le serveur **CS-02** avec des identifiants d'administration du domaine Abdel.lan, puis dans le Gestionnaire de serveur, nous avons cliqué sur **Ajouter des rôles et des fonctionnalités**.
- **Étape 2** : cliquer sur **Suivant** pour passer les pages **Avant de commencer**, **Sélectionner le type d'installation** et **Sélectionner le serveur de destination**.

- **Étape 3** : dans l'étape **Sélectionner des rôles de serveurs**, nous avons coché la case correspondant au rôle **Services de certificats Active Directory**, puis cliqué sur le bouton **Ajouter des fonctionnalités** ensuite sur **Suivant**.
- **Étape 4** : dans l'étape **Sélectionner des fonctionnalités**, nous avons cliqué sur **Suivant**.
- **Étape 5** : dans l'étape **Services de certificats Active Directory**, nous avons cliqué sur **Suivant**.
- **Étape 6** : dans l'étape **Sélectionner des services de rôle**, nous avons coché la case **Autorité de certification**, puis **Inscription de l'autorité de certification via le Web,** cette dernière fourni une interface Web simple nous permettant d'effectuer des taches telle que la demande et le renouvellement de Certificat, la récupération des listes de révocation de certificats (CRL) etc…puis nous avons cliqué sur **Ajouter des fonctionnalités**.

  Le fait que nous avons ajouté le service de rôle **Inscription de l'autorité de certification via le web**, l'assistant propose alors l'installation du rôle **serveur web (IIS)**. Puis nous avons cliqué sur **Suivant** :

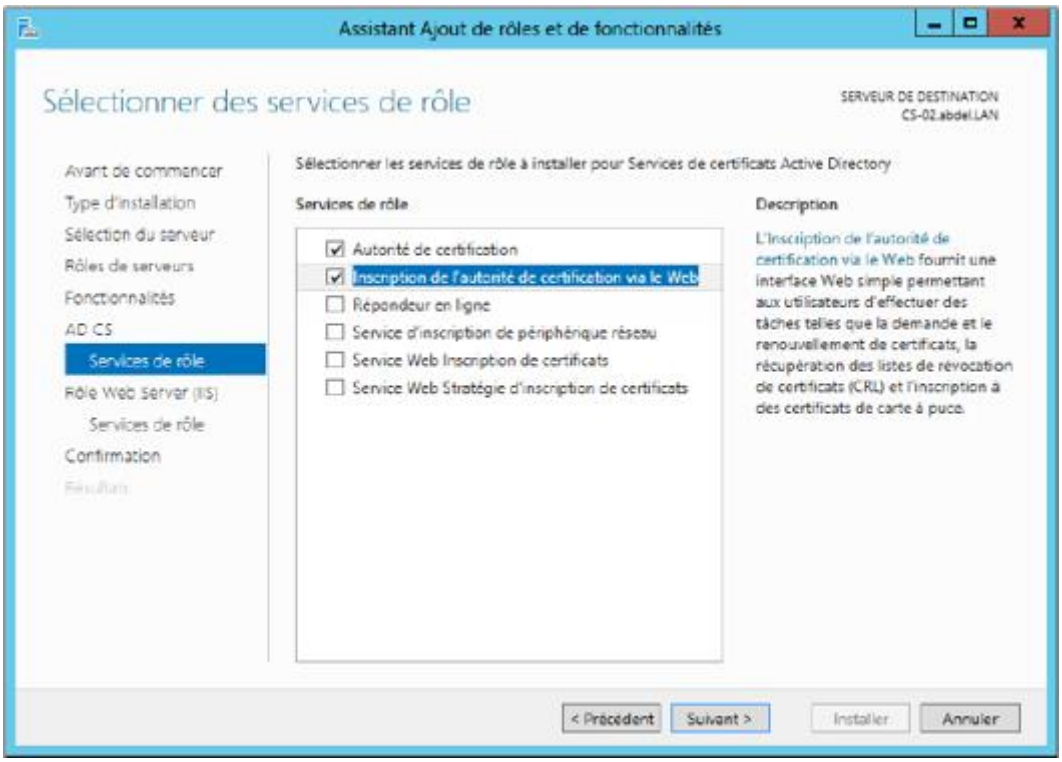


Figure IV.18. Assistant d'ajout des rôles et fonctionnalités.

- **Étape 7** : Dans l'étape **Rôle Web Server (IIS)**, nous avons cliqué sur **Suivant**.
- **Étape 8** : Dans l'étape **Sélectionner des services de rôle**, nous avons cliqué sur **Suivant**.

- **Étape 9** : Dans l'étape **Confirmer les sélections d'installation**, nous avons cliqué sur **Installer**.
- **Étape 10** : Dans l'étape **Progression de l'installation**, nous avons cliqué sur **Configurer les services de certificats Active Directory sur le serveur de destination**.
- **Étape 11** : dans l'étape **Informations d'identification**, nous devons nous assurer qu'un compte d'administration du domaine ABDEL.LAN a été renseigné, puis nous avons cliqué sur **Suivant**.
- **Étape 12** : dans l'étape **Services de rôle**, nous avons coché les cases associées aux rôles **Autorité de certification** et **Inscription de l'autorité de certification via le Web**. Ensuite nous avons cliqué sur **Suivant** :

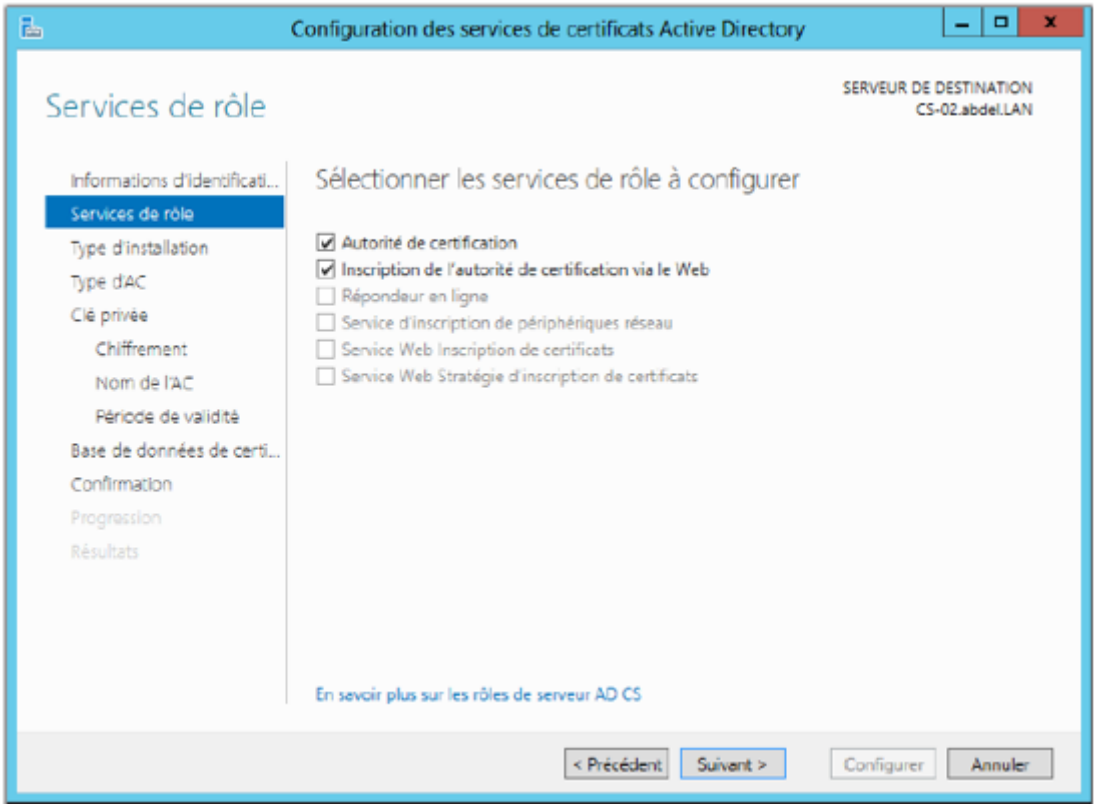


Figure IV.19. Configuration des services de certificats Active Directory.

- **Étape 13** : Dans l'étape **Type d'installation**, nous avons coché la case **Autorité de certification d'entreprise** puis cliqué sur **Suivant**.
- **Étape 14** : Dans l'étape **Type d'autorité de certification**, nous avons coché la case **Autorité de certification secondaire** puis cliqué sur **Suivant**.
- **Étape 15** : Dans l'étape **Clé privée**, nous avons coché la case **Créer une clé privée**, puis cliqué sur **Suivant**.

- **Étape 16** : Dans l'étape **Chiffrement pour l'autorité de certification**, laisser les options par défaut et cliquez sur **Suivant**.
- **Étape 17** : Dans l'étape **Nom de l'autorité de certification**, nous avons tapé *ABDEL-Emettrice-CA* dans le champ **Nom commun de cette AC**, ce nom est ajouté à tous les certificats émis par l'autorité de certification dont les valeurs des suffixes du nom unique sont généré automatiquement mais modifiable, ensuite nous avons cliqué sur **Suivant** :

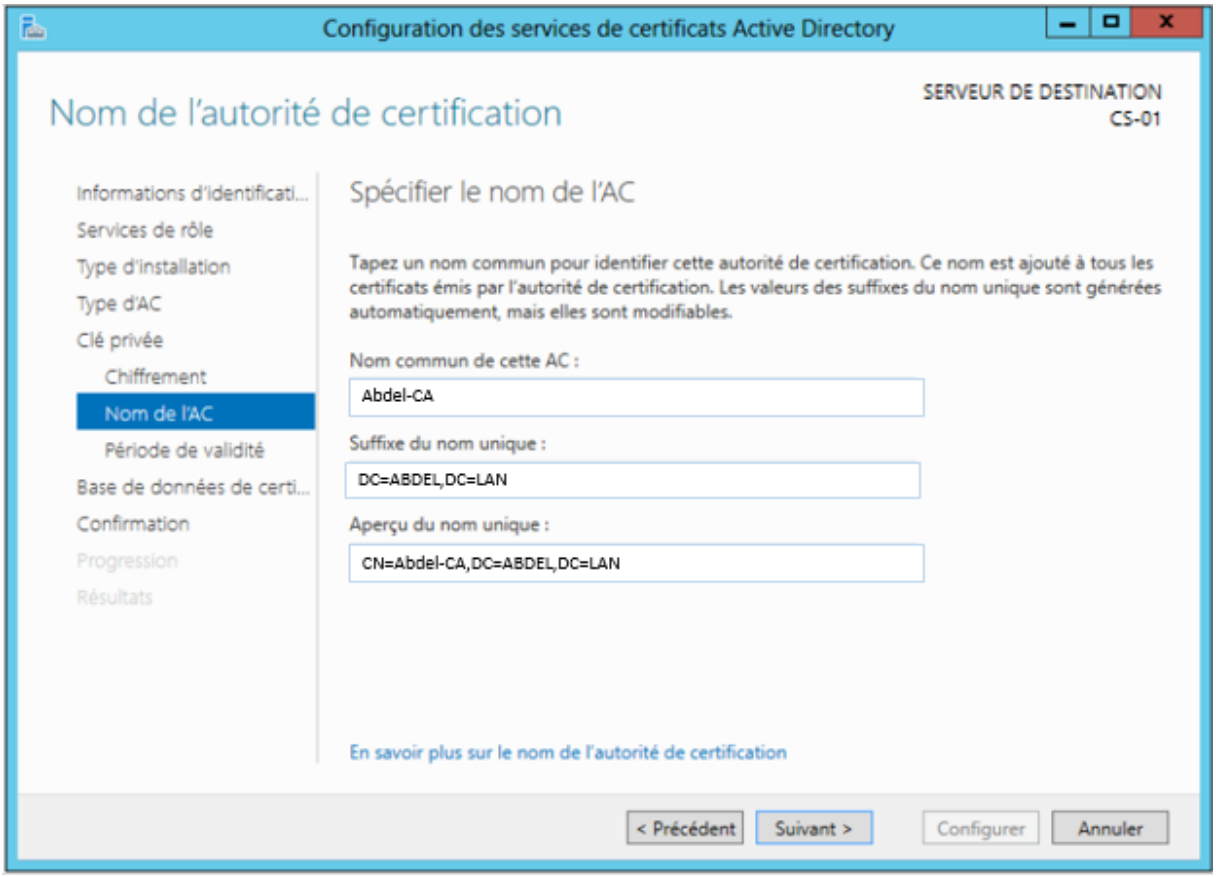


Figure IV.20. Configuration des services de certificats AD.

- **Étape 18** : Dans cette étape nous avons demandé un certificat aux prés d'une autorité parente et ce afin de permettre à cette autorité de certification secondaire d'émettre des certificats. Pour cela nous avons coché la case **Enregistrer une demande de certificat dans un fichier de l'ordinateur cible** avec l'identification de l'emplacement d'enregistrement du fichier de demande de certificat ensuite nous avons cliqué sur **Suivant.**

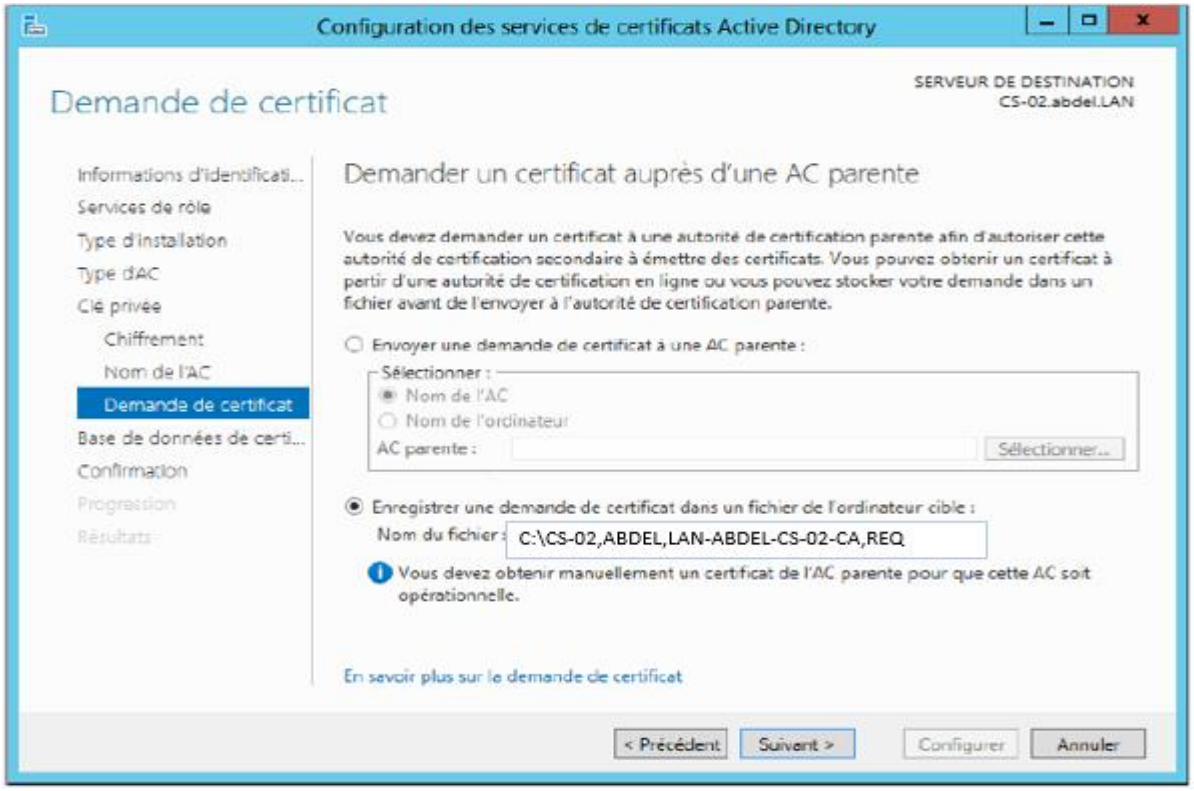


Figure IV.21. Emplacement des bases de données.

- **Étape 19** : dans l'étape **Base de données de l'autorité de certification**, nous avons spécifié le répertoire *D:\CA\CertDB* comme emplacement de la base de données de certificats dans, et le répertoire *E:\CA\CertLog* comme emplacement du journal de la base de données de certificats ensuite nous avons cliqué sur **Suivant.**

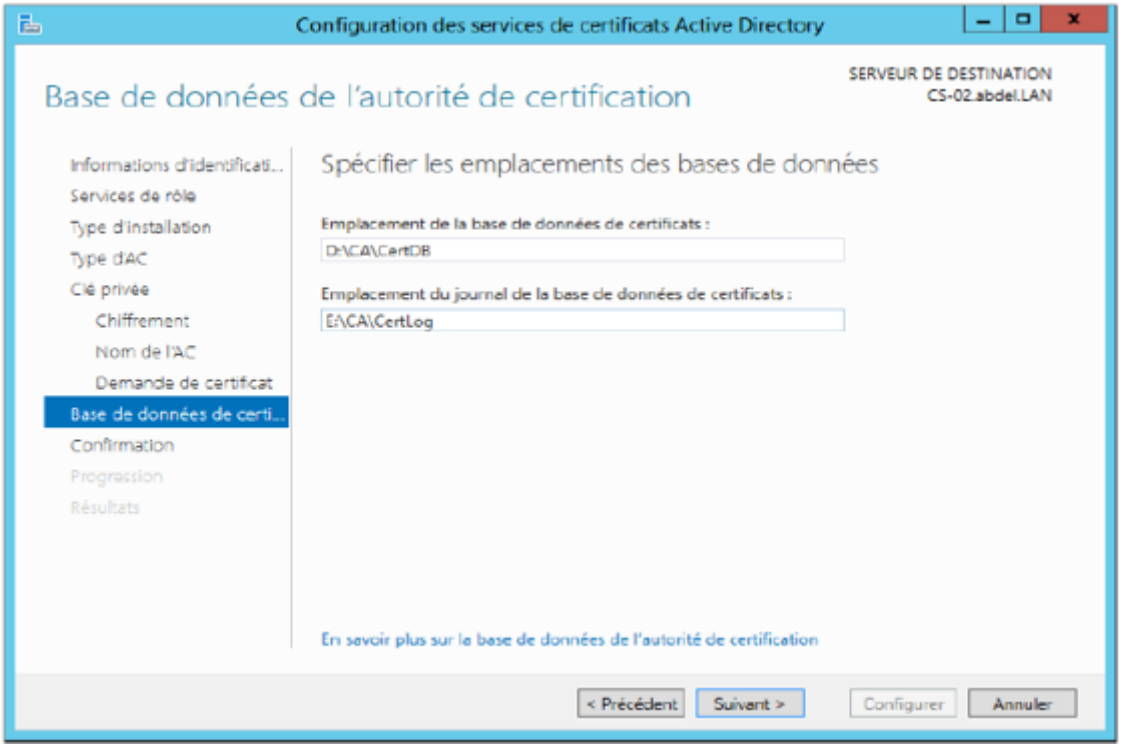


Figure IV.22. Emplacement des bases de données.

- **Étape 20** : Dans l'étape **Confirmation**, nous avons vérifié les informations de configuration de notre autorité de certification secondaire, ensuite nous avons cliqué sur **Configurer**.
- **Étape 21** : Dans l'étape **Résultats**, l'installation de l'autorité de certification d'entreprise affiche un message d'avertissement, indiquant qu'il ne faut pas oublier de demander un certificat de l'autorité de certification parente pour autoriser cette autorité de certification d'entreprise à émettre des certificats.

Nous nous sommes assuré que l'installation du service de rôle **Inscription de l'autorité de certification via le web** indique un état de configuration réussie. Ensuite nous avons cliqué sur **Fermer**.

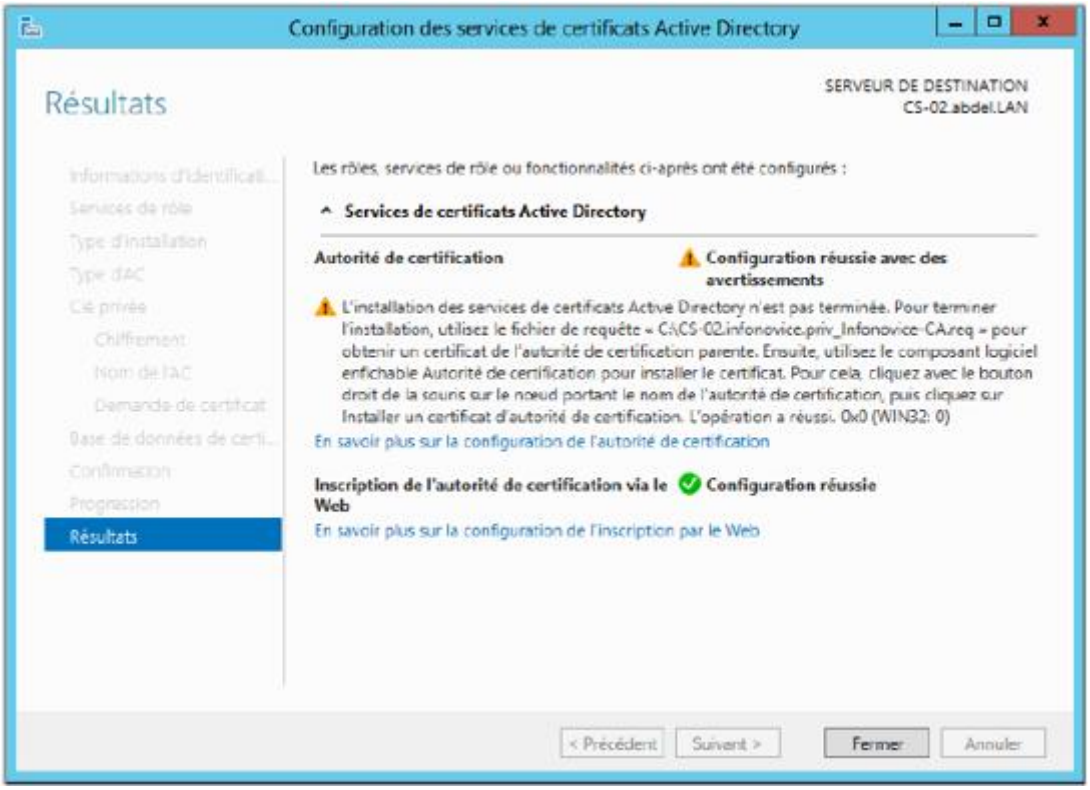


Figure IV.23. Résultat de la configuration.

- ✓ À ce stade, l'autorité de certification d'entreprise émettrice est installée et configurée, mais ne fonctionne pas. Il faut, par la suite, utiliser le fichier de requête généré dans le processus d'installation, pour obtenir un certificat de l'autorité de certification parente.

IV.6.2. Activation de la CA Emettrice

Ces étapes permettent d'activer une CA émettrice hébergeant le rôle de serveur AD CS en tant qu'autorité de certification d'entreprise sur le serveur **CS-02**.

- ➢ **Etape1 :** nous avons ouvert une session sur le serveur **CS-02** avec des identifiants d'administration, puis nous avons démarré la console de gestion **Autorité de certification**. avec un clic droit sur l'autorité de certification **abdel-Root-CA** et un cliqué sur **Toutes les tâches** puis **Soumettre une nouvelle demande.**

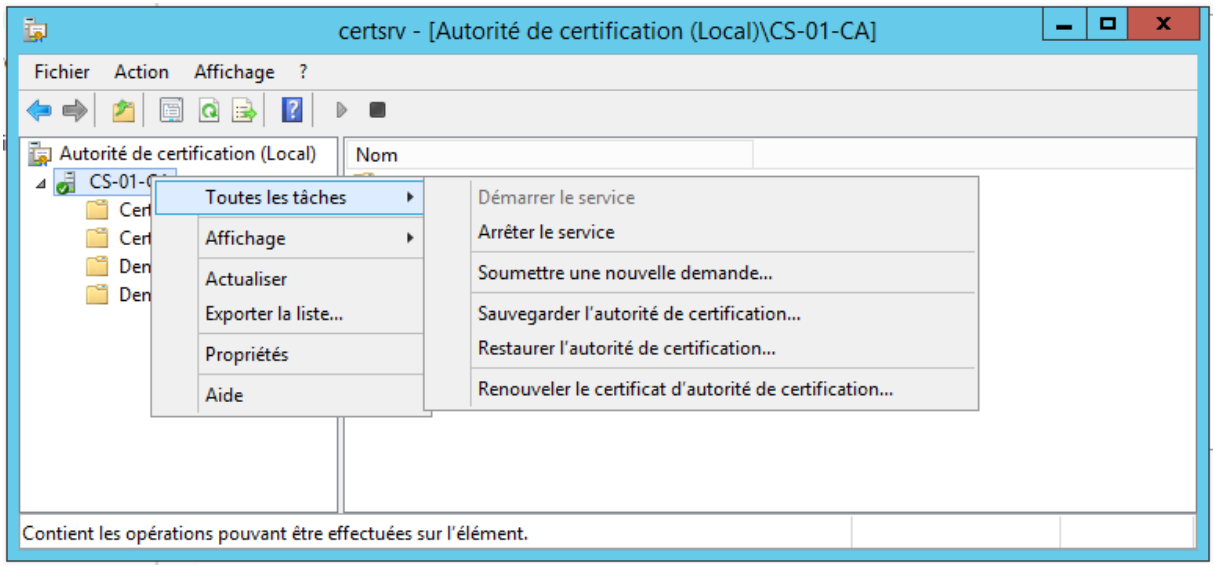


Figure IV.24. Soumettre une nouvelle demande.

- **Étape 2** : Dans l'arborescence de la console certsrv, nous avons sélectionné le répertoire **Demandes en attente**. puis un clic droit sur la requête disponible, ensuite un cliqué sur **Toutes les tâches**, puis sur **Délivrer**

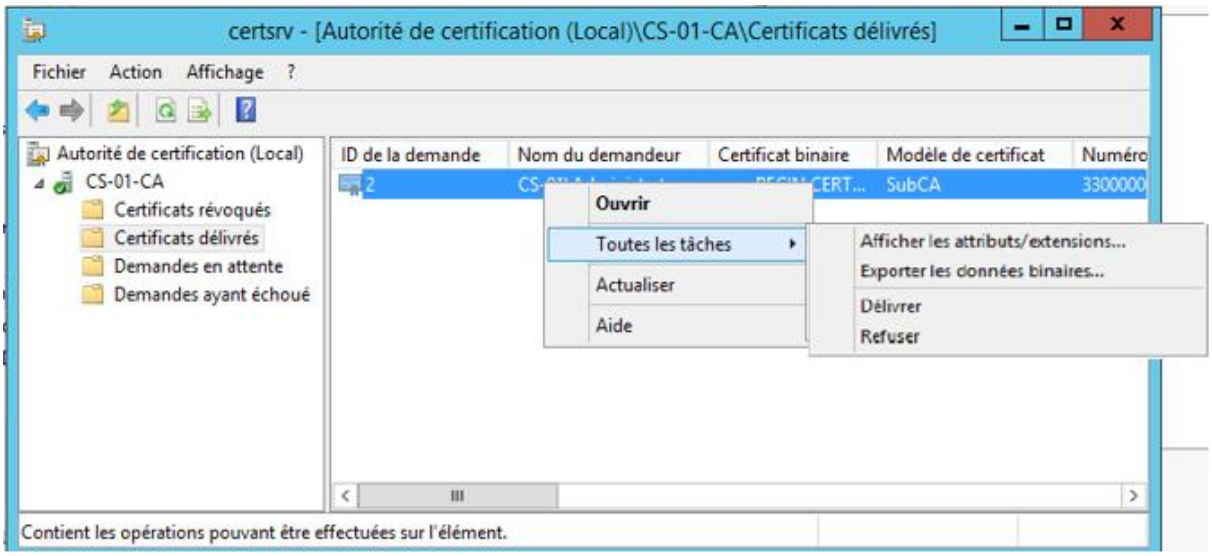


Figure IV.25. Délivrer un certificat.

- **Etape 3 :**Dans l'étape **Format du fichier d'exportation**, nous avons coché la case **Standard de syntaxe de message cryptographique - Certificats PKCS #7(.P7B)** ainsi que la case **Inclure tous les certificats dans le chemin d'accès de certification, si possible** en fin sur **Suivant.**

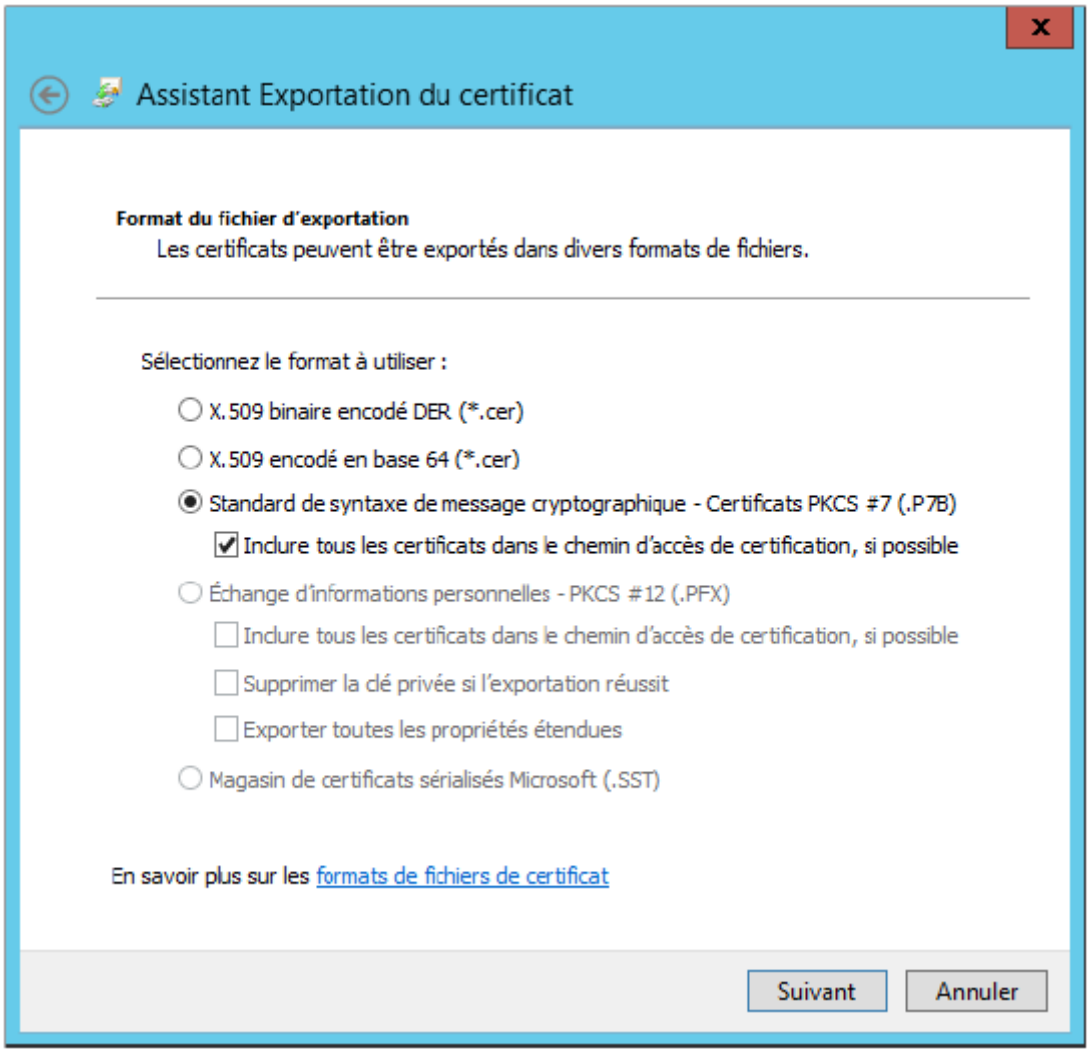


Figure IV.26. Format du fichier.

- **Étape 4** : Dans l'étape **Fichier à exporter**, nous avons cliqué sur **Parcourir**. Nous avons navigué dans le répertoire partagé \\CS-02\Partage et tapé*CA-Intermediaire.p7b* dans le **Nom du fichier**. Ensuite nous avons cliqué sur **Enregistrer**.
- **Etape5 :** Nous avons ouvert une session sur le serveur **CS-02** avec des identifiants d'administration du domaine abdel.lan, et démarré la console de gestion **Autorité de certification**. Ensuite nous avons fait un clic droit sur l'autorité de certification *abdel-Emettrice-CA* et cliqué sur **Toutes les tâches** puis **Installer un certificat d'autorité de certification.**

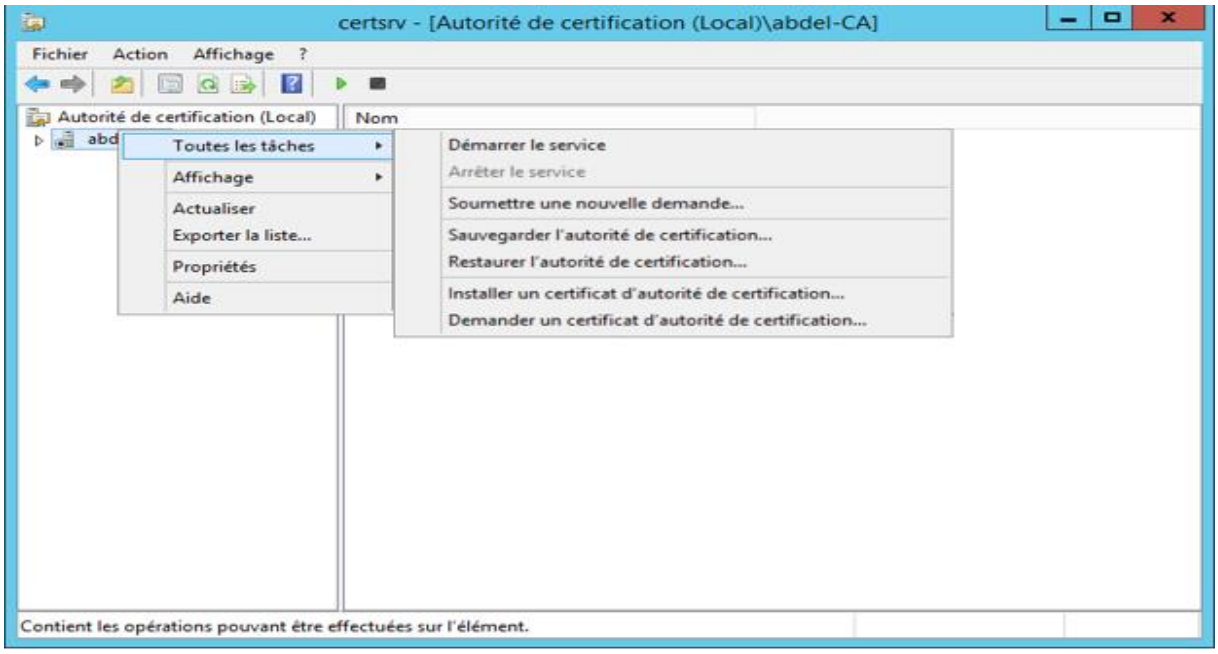


Figure IV.27. Installation d'un certificat.

- **Etape 5 :** Dans la console de gestion certSrv sur le serveur **CS-02**, nous avons fait un clic droit sur l'autorité de certification, puis cliqué sur **Toutes les tâches** et **Démarrer le service**.

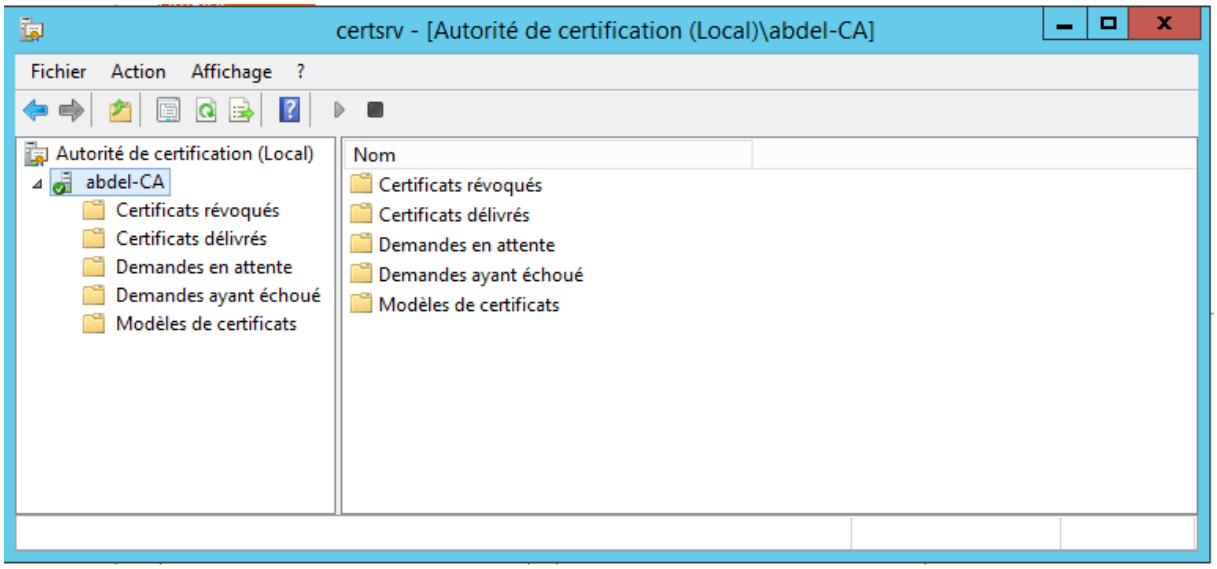


Figure IV.28. Démarrer le service.

IV.7. Publication d'un certificat via des GPO

Cette partie va nous permettre de publier le certificat de l'autorité de certification racine de l'infrastructure PKI, afin que l'ensemble des postes membres du domaine puisse approuver le certificat émis.

- **Étape 1** : Dans cette étape nous avons ouvert une session sur le serveur **DC-01** avec des identifiants d'administration du domaine abdel.lan, puis dans le Gestionnaire de serveur, nous avons cliqué sur **Outils**, puis sur **Gestion de stratégies de groupe**.
- **Étape 2** : Dans cette étape nous avons développé l'arborescence de la console, puis édité le GPO **Default Domain Policy**, sous le domaine abdel.lan.
- **Étape 3** : Dans cette étape nous avons développé l'arborescence de la console et sélectionné le répertoire suivant :

Configuration ordinateur - Stratégies - Paramètres Windows - Paramètres de sécurité – Stratégie de clé publique - Autorité de certification racine de confiance.

- **Étape 4** : Dans cette étape nous avons fait un clic droit sur **Autorité de certification racine de confiance**, puis cliqué sur **Importer**.
- **Étape 5** : Dans la fenêtre **Assistant importation du certificat**, nous avons cliqué sur **Suivant** à l'écran de bienvenue.
- **Étape 6** : Dans l'étape **Fichiers à importer**, nous avons cliqué sur **Parcourir**.
- **Étape 7** : Dans cette étape nous avons sélectionné le fichier **\\CS-02\Partage\Certificat-Root.cer** et cliqué sur **Ouvrir**, puis **Suivant**.
- **Étape 8** : Dans l'étape **Magasin de certificats**, nous nous sommes assuré que le magasin sélectionné est bien **Autorité de certification racine de confiance**, puis cliqué sur **Suivant**.
- **Étape 9** : Ensuite nous avons cliqué sur **Terminer** puis **OK**.
- **Étape 10** : En fin nous avons cliqué sur fermer l'éditeur ainsi que la console de Gestion de stratégies de groupe.

IV.8. Configurer l'interface Web

Ces étapes vont nous permettre de configurer l'interface web des services de certificat afin d'utiliser le protocole HTTPS.

- **Étape 1** : Dans cette étape nous avons ouvert une session sur le serveur **CS-02**, avec des identifiants d'administration du domaine abdel.lan. Dans le Gestionnaire de serveur, nous avons cliqué sur **Outils**, puis sur **Gestionnaire des services Internet (IIS)**.
- **Étape 2** : Dans cette étape nous avons sélectionné le nom du serveur hébergeant l'autorité de certification d'entreprise émettrice se trouvant dans la partie gauche de l'arborescence de la console. Ensuite nous avons fait un double clic sur **Certificats de serveur** situé dans la partie centrale,

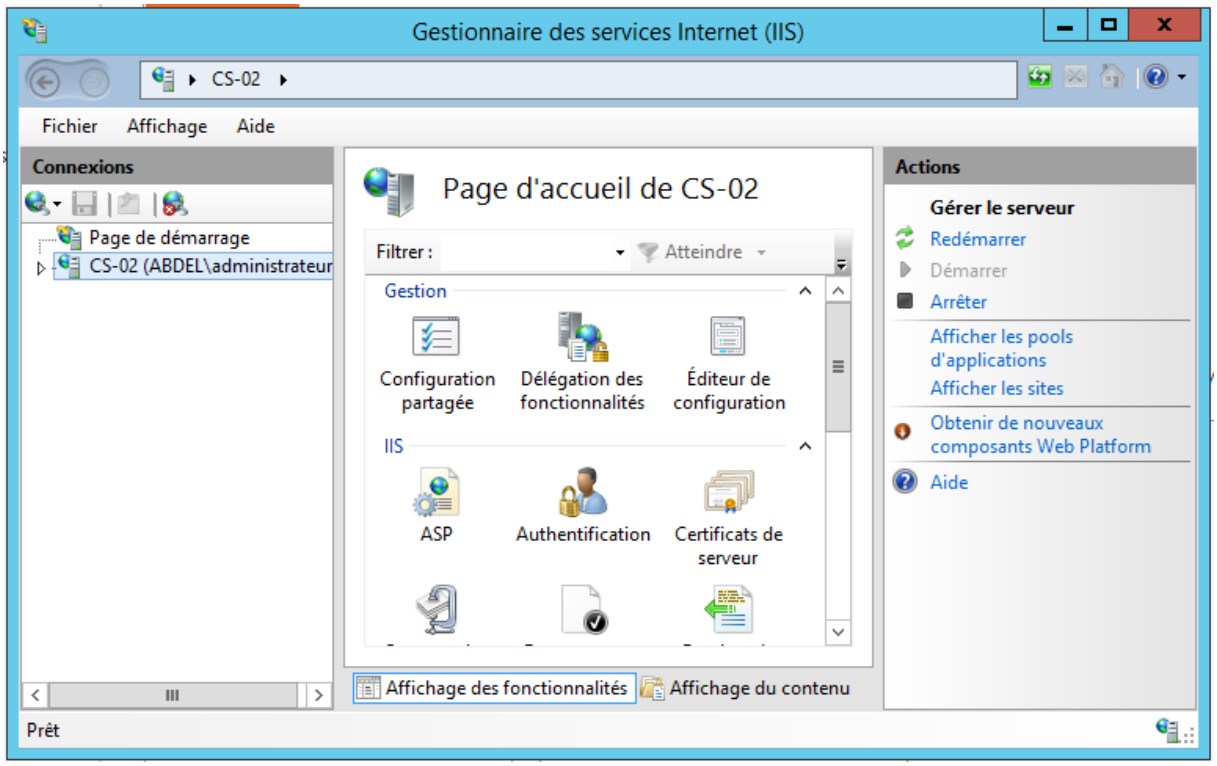


Figure IV.29. Page d'accueil gestionnaire des services IIS.

- **Étape 3** : Dans la partie droite de la console, nous avons cliqué sur **Créer une demande de certificat**

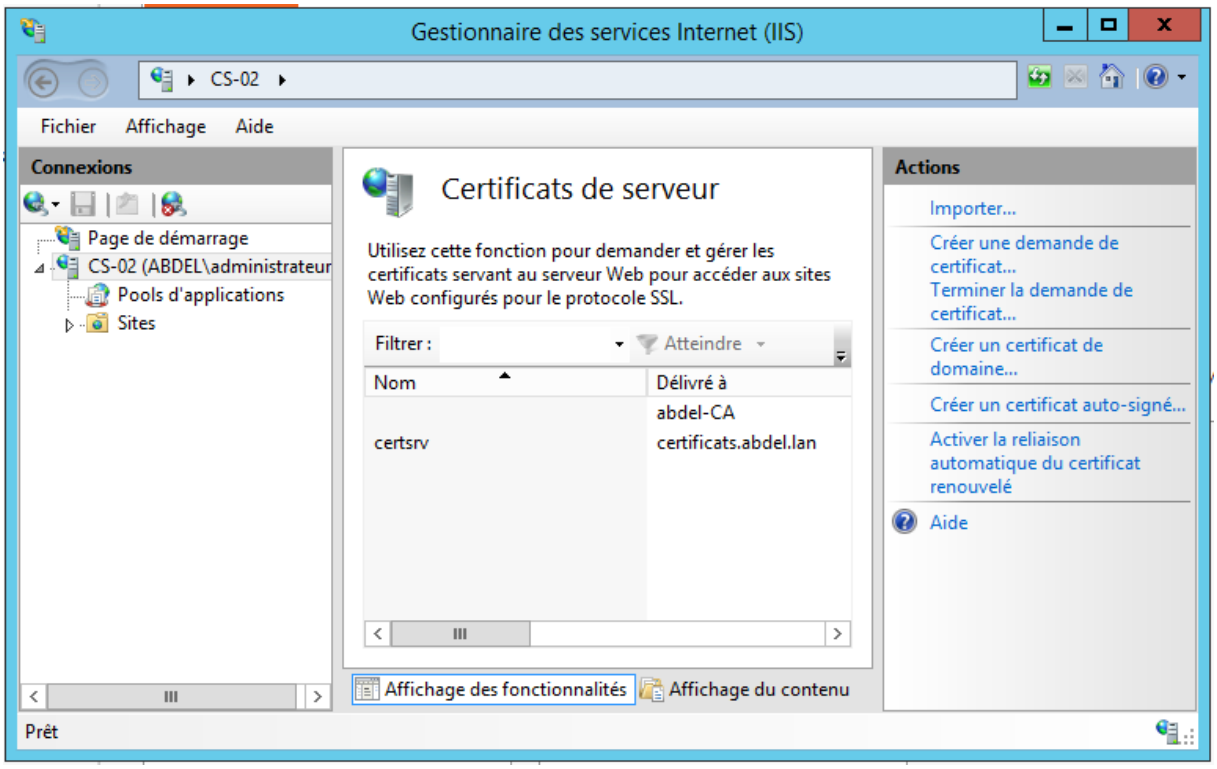


Figure IV.30. Créer une demande de certificat.

➢ **Étape 4** : Dans la fenêtre **Demande de certificat**, nous avons remplis les informations suivantes, puis cliqué sur **Suivant.**

- **Nom commun** : certificats.abdel.lan
- **Organisation** : abdel
- **Unité d'organisation** : CERTIFICATS
- **Ville** : Tizi-Ouzou
- **Département/région** : bastos
- **Pays/région** : DZ

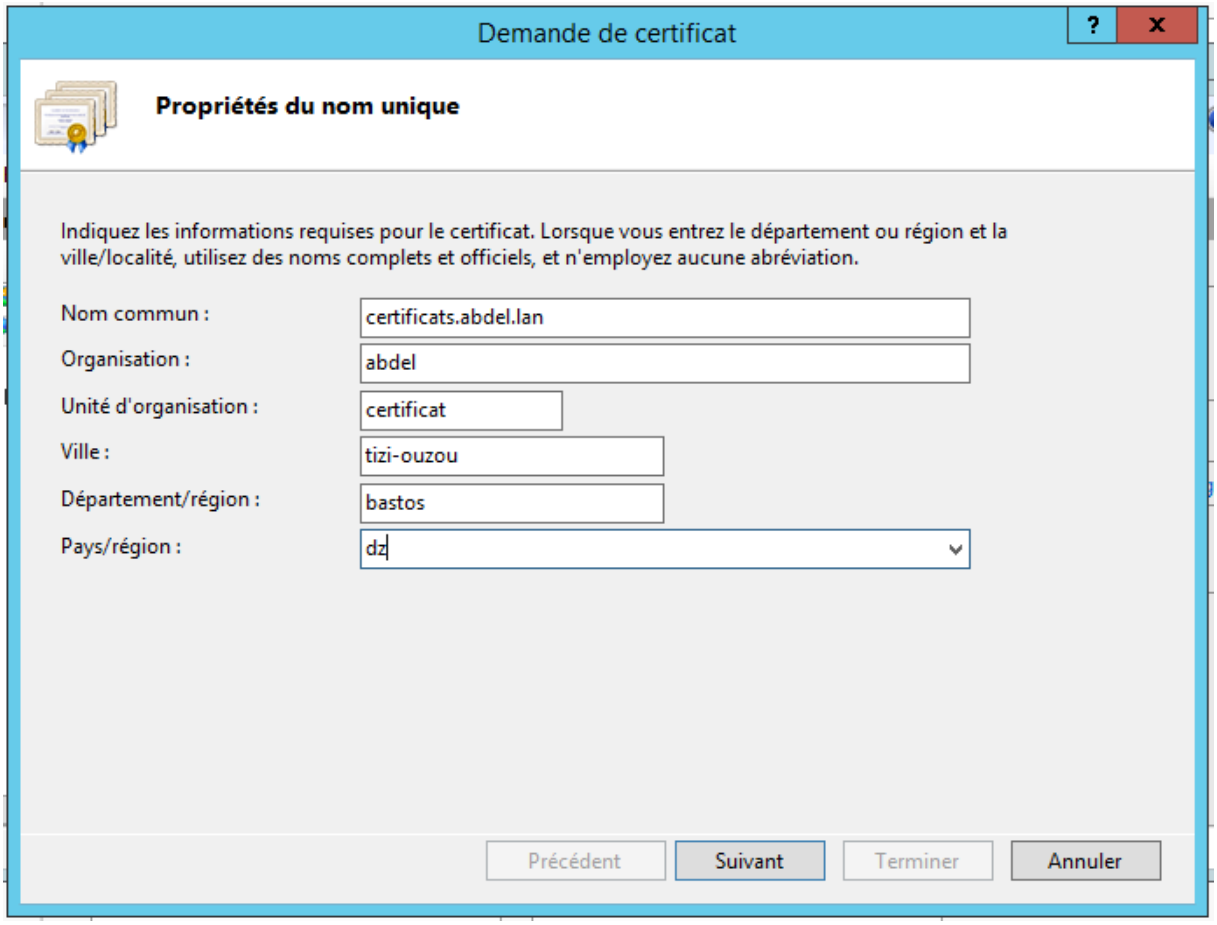


Figure IV.31. Demande de certificat.

❖ *Le champ **Nom commun** correspond à l'URL définitive, soit http://certificats.abdel.lan.*

➢ **Étape 5** : Cette étape **Propriétés du fournisseur de services de chiffrement**, consiste a sélectionner un fournisseur de services de chiffrement et une longueur en bit. La longueur en bit de la clé de chiffrement détermine l'efficacité de chiffrement, dans notre cas nous avons choisi une longueur en bit de 2048, ensuite nous avons cliqué sur **Suivant**.

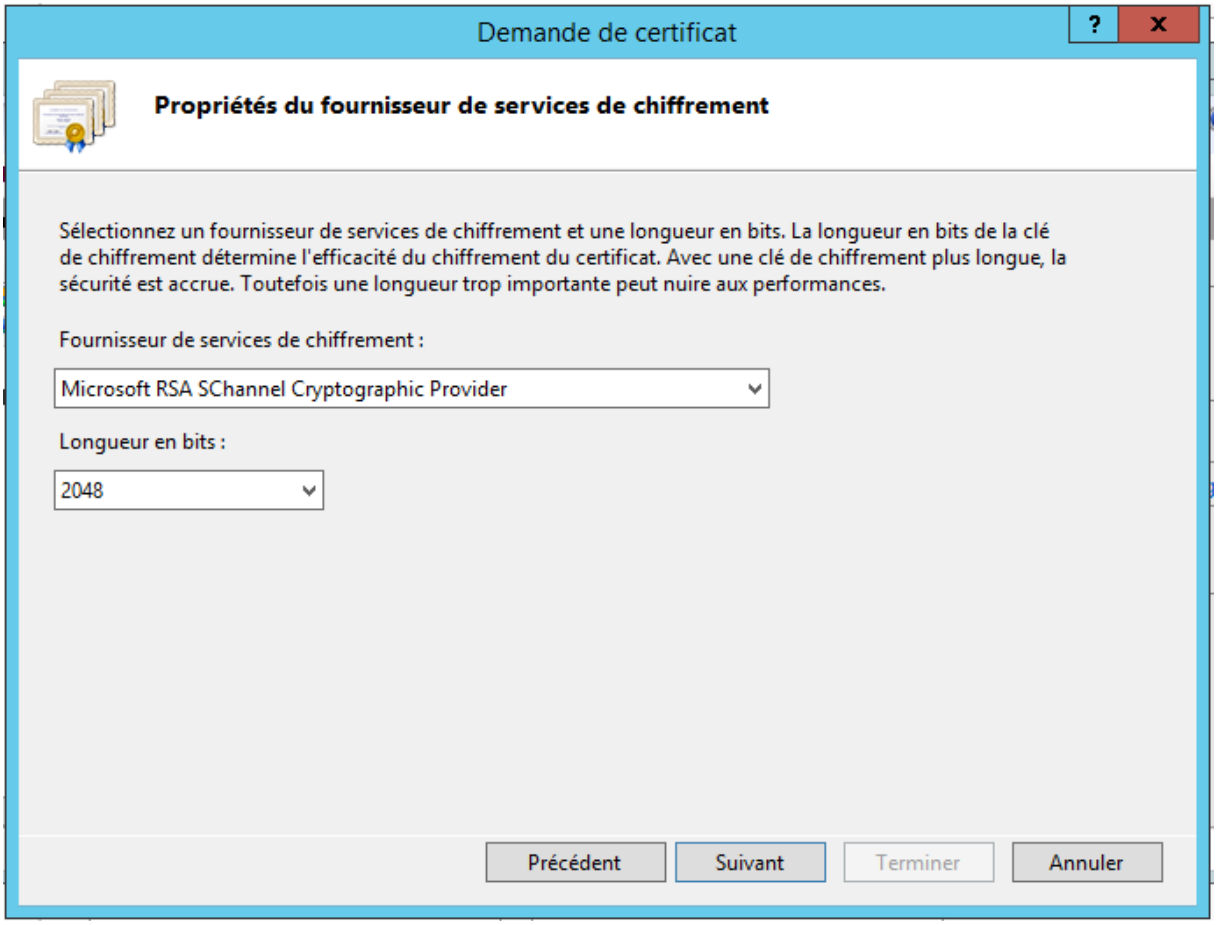


Figure IV.32. Choix de la longueur en bit.

- **Étape 6** : Nous avons créé le répertoire C:\Certificats. Puis dans l'étape **Nom du fichier**, nous avons le nom du fichier pour la demande de certificat suivant : *C:\Certificats\Demande.txt*, ensuite cliqué sur **Terminer.**

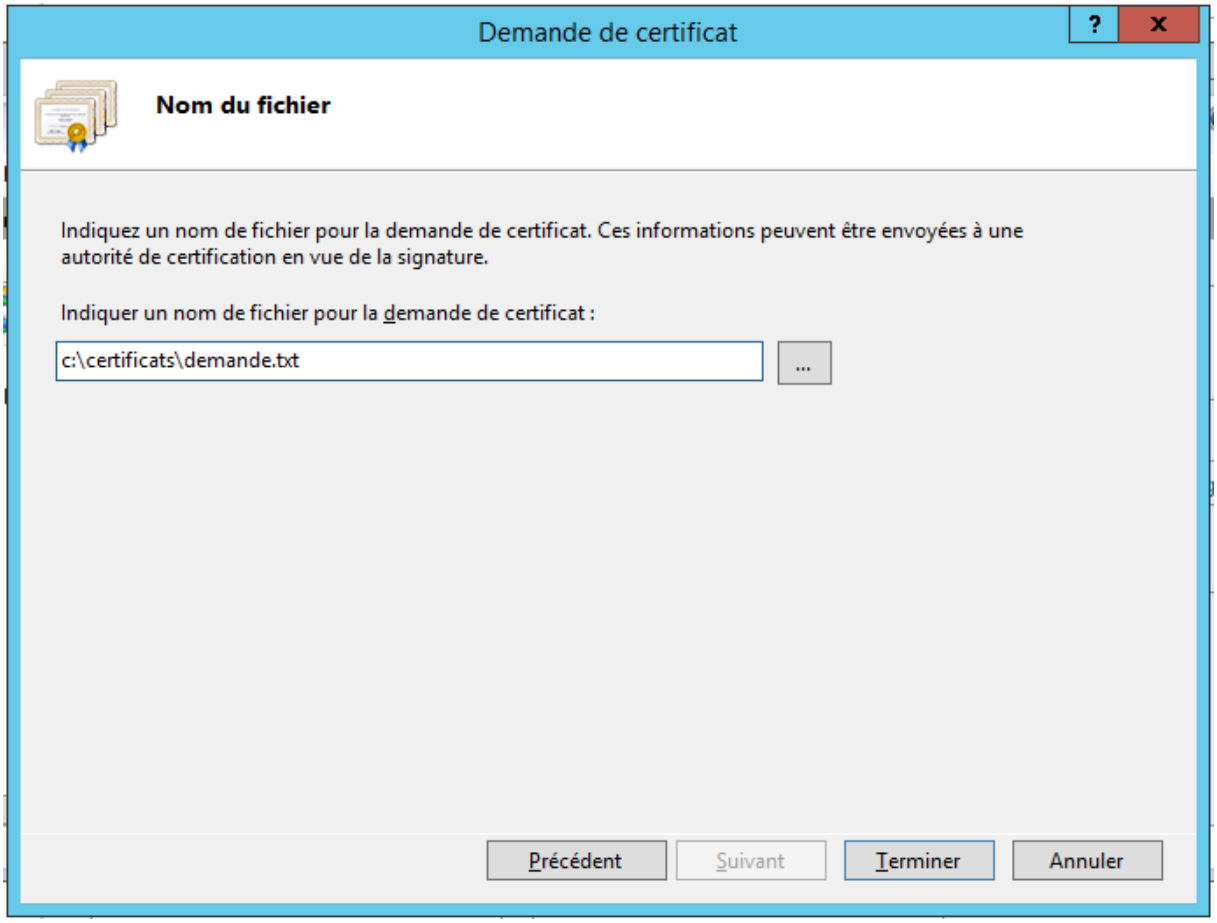


Figure IV.33. Nom du fichier.

- **Étape 7** : dans cette étape nous avons ouvert le navigateur Internet Explorer et nous avons tapé l'URL : *https://certificats.abdel.lan/certsrv/*, ce site va nous permettre de demander des certificats ou d'affiché le statut d'une requête de certificat en attente et de télécharger un certificat d'autorité de certification, une chaine de certification ou une liste de révocation des certificats. Dans notre cas nous avons cliqué sur **Demander un certificat.**

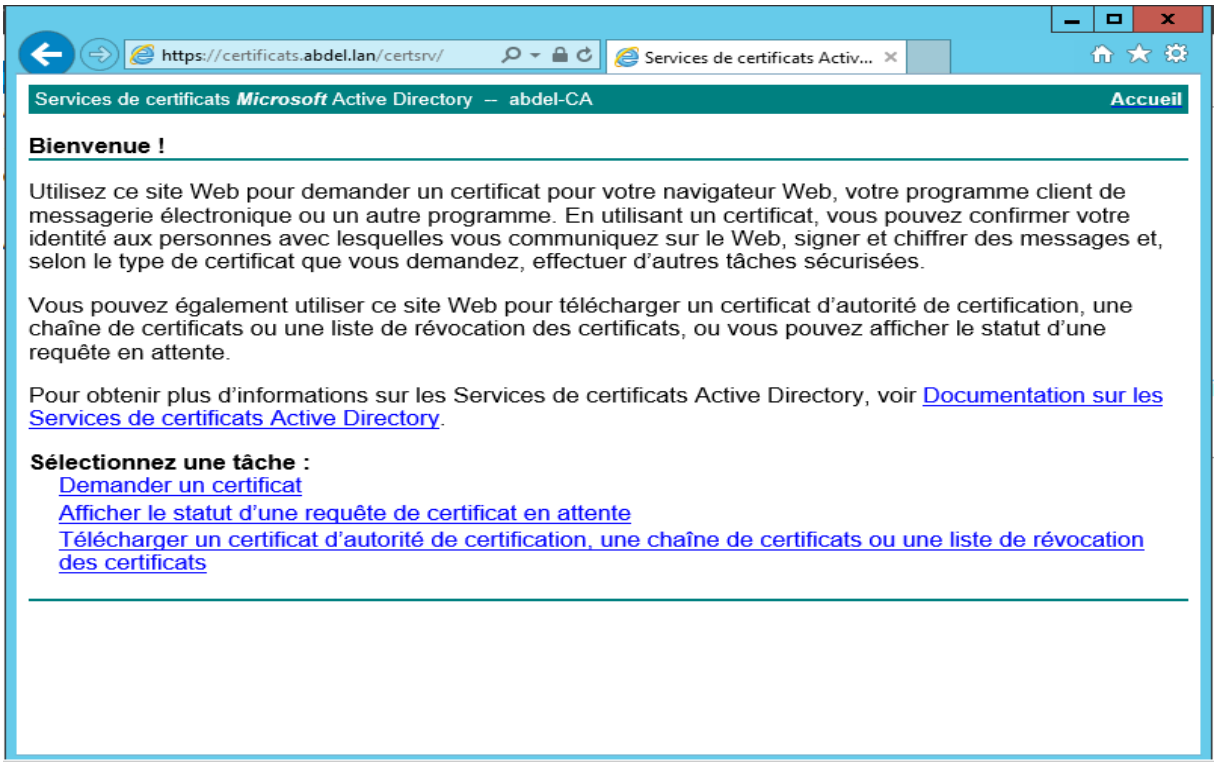


Figure IV.34. Demande de certificat via le site Web.

- **Étape 8** : Dans cette étape nous avons cliqué sur **demande de certificat avancée.**

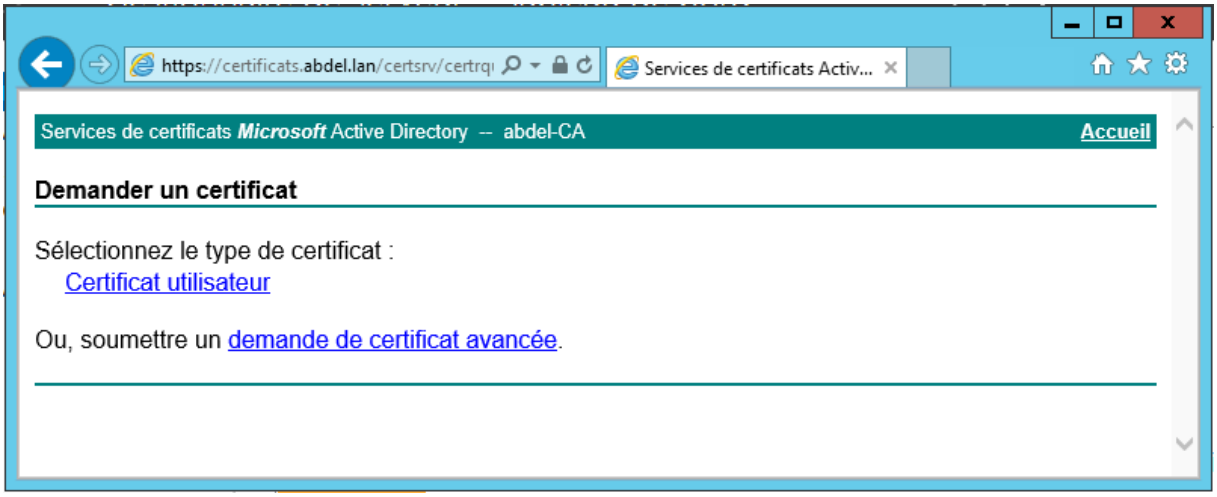


Figure IV.35. Demande de certificat avancée.

- **Étape 9** : Dans cette étape la stratégie de l'autorité de certification détermine le type de certificat que nous pourrons demander, dans notre cas nous avons cliqué sur **Soumettez une demande de certificat en utilisant un fichier.**

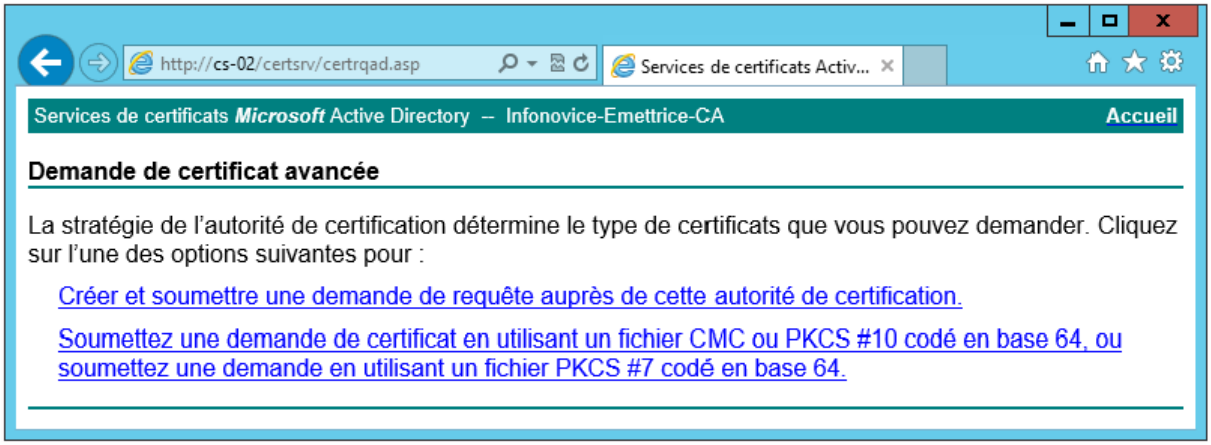


Figure IV.36. Soumettre une demande.

- **Étape 10** : Dans cette étape nous avons copié le contenu du fichier *C:\Certificats\Demande.txt* dans la section **Demande enregistrée**. Puis nous avons sélectionné le modèle de certificat **Serveur Web**, ensuite cliqué sur **Envoyer**.

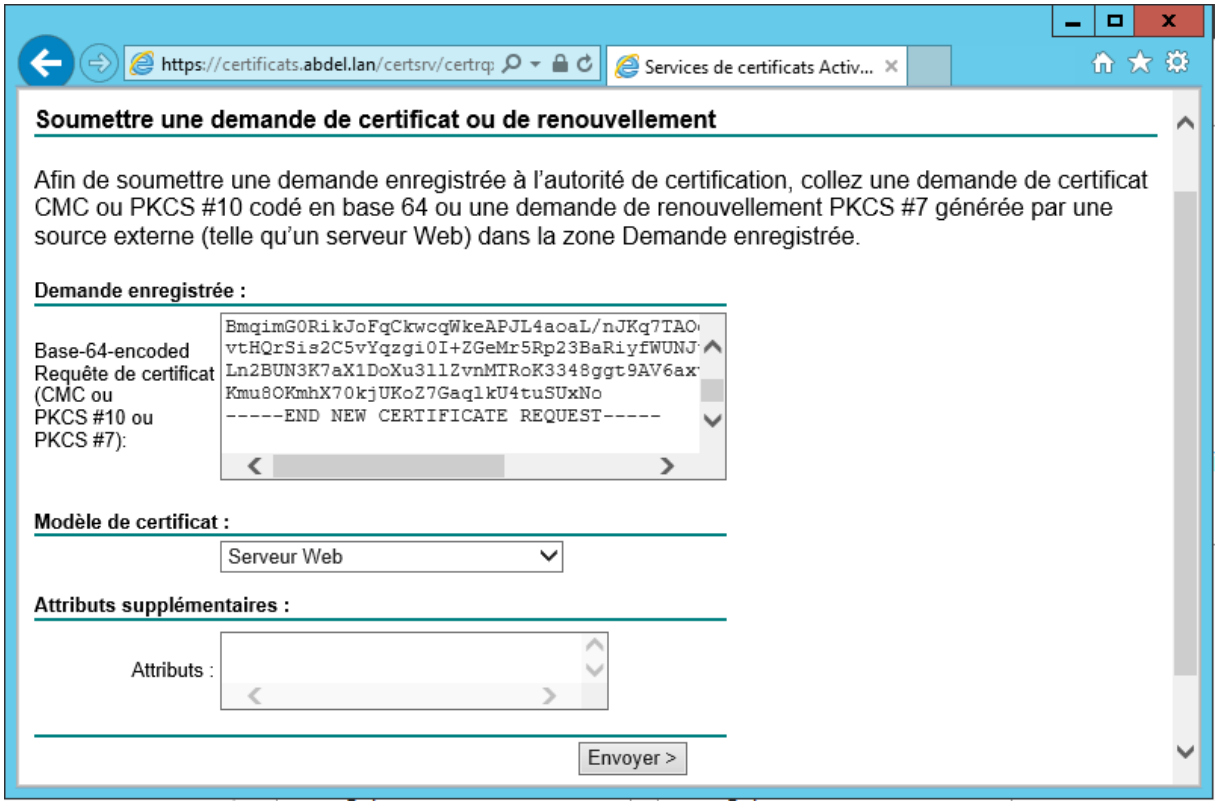


Figure IV.37. Modèle de certificat à émettre.

- **Étape 11** : Dans cette étape nous avons cliqué sur **Télécharger le certificat** et enregistrer le fichier *Certnew.cer* dans le répertoire *C:\Certificats*.

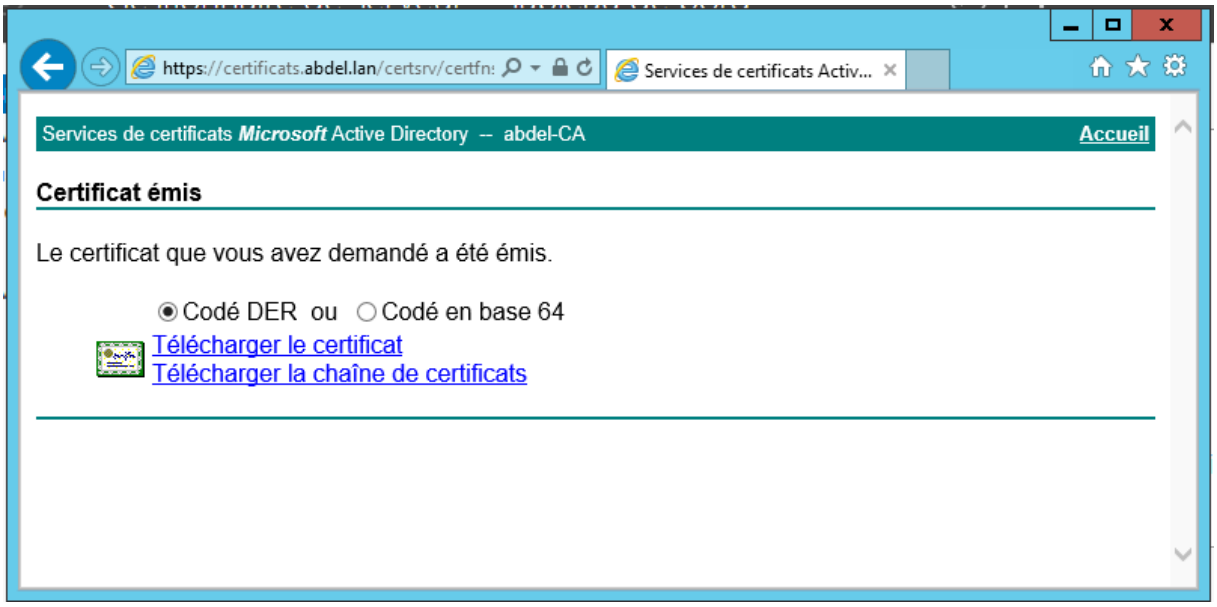


Figure IV.38. Emettre le certificat.

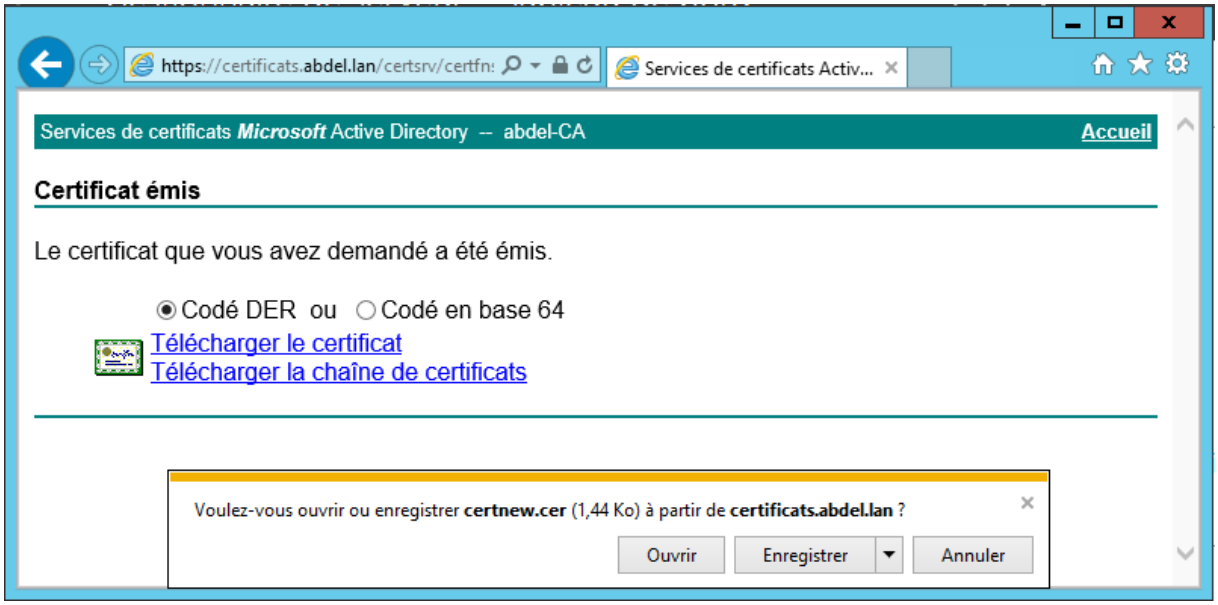


Figure IV.39. Enregistrement du fichier Certnew.cer.

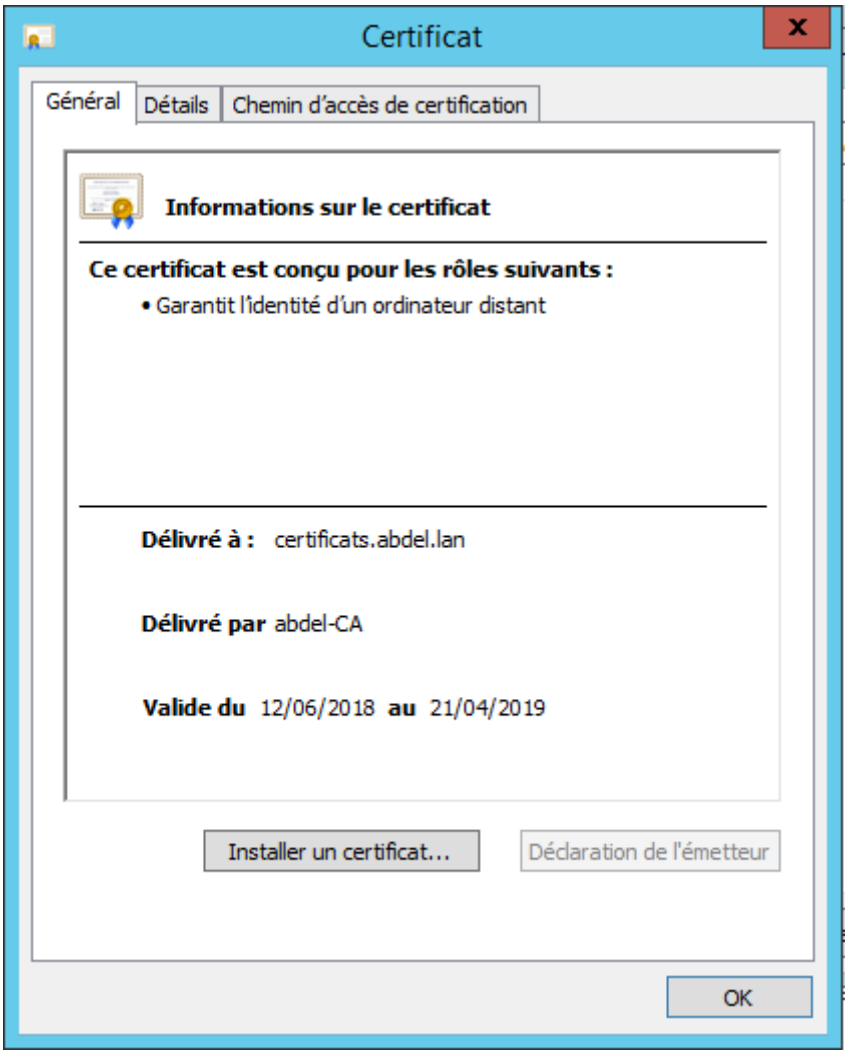


Figure IV.40. Information sur le certificat.

- **Étape 12** : Dans cette étape nous avons basculé sur la console de gestion IIS et nous avons cliqué sur **Terminer la demande de certificat** se trouvant dans la partie droite de la console.
- **Étape 13** : Dans l'étape **Indiquer la réponse de l'autorité de certification**, nous avons sélectionné l'emplacement du fichier de réponse *C:\Certificats\certnew.cer*, et tapé *certsrv* dans le champ **Nom convivial**, ensuite nous avons sélectionné **Personnel** dans le champ **Sélectionnez un magasin de certificats pour le nouveau certificat** et Cliqué ensuite sur **OK**.
- **Étape 14** : Dans cette étape nous avons basculé sur la console de gestion IIS, puis développé l'arborescence et sélectionné **Default Web Site**. Dans la partie droite, nous avons cliqué sur **Liaison de site**.
- **Étape 15** : Dans la fenêtre **liaisons de sites**, nous avons cliqué sur **Ajouter**.

- **Étape 16** : Dans la fenêtre **Ajouter la liaison de site**, nous avons entré les informations suivantes et cliqué sur **OK** puis sur **Fermer**.

- **Type** : https
- **Port** : 443
- **Nom de l'hôte** : certificats.abdel.lan
- **Certificat SSL** : certsrv

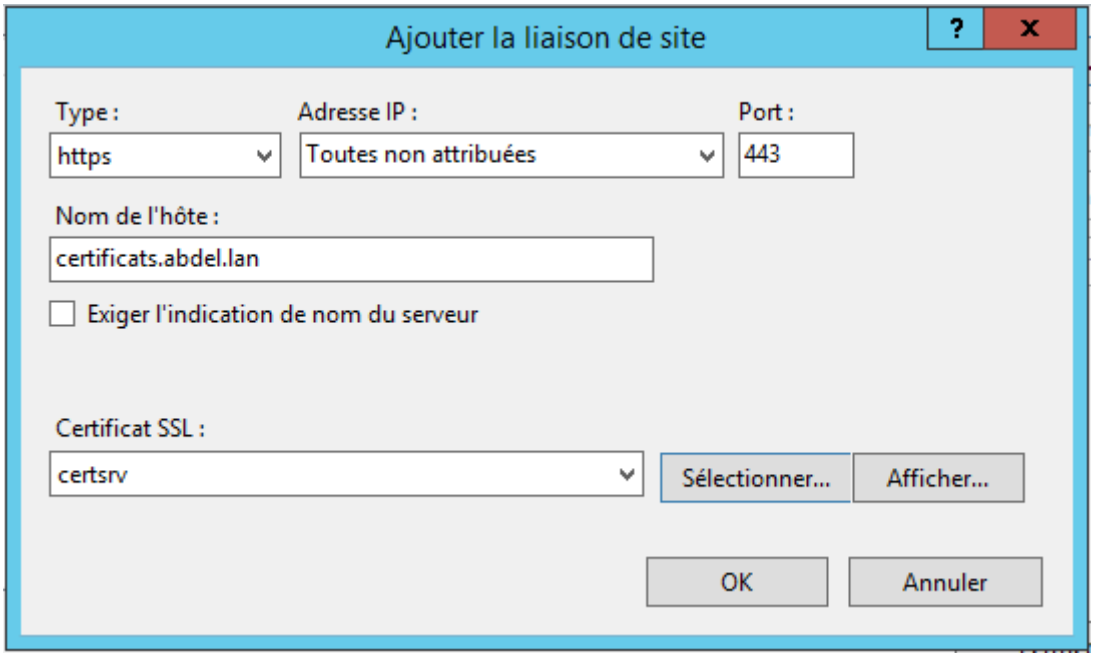


Figure IV.41. Ajouter la liaison du site.

- **Étape 17** : Dans cette étape nous avons développé l'arborescence de Default Web Site et sélectionné le répertoire virtuel certsrv. Dans la partie centrale, nous avons fait un double clic sur **Paramètres SSL**.
- **Étape 18** : Dans cette étape nous avons coché la case **Exiger SSL** et cliqué sur **Appliquer**.
- **Étape 19** : Sur le serveur **DC-01**, nous avons ouvert le Gestionnaire de serveur, et cliqué sur **Outils** puis sur **DNS**.
- **Étape 20** : Dans cette étape nous avons développé l'arborescence de la console, puis créer un nouvel enregistrement de ressources *CNAME* dans la **Zone de recherche directe abdel.lan**. Nous avons indiqué les informations suivantes et cliqué sur **OK** :

- **Nom de l'alias** : certificats
- **Nom de domaine pleinement qualifié (FQDN)** : certificats.abdel.lan
- **Nom de domaine complet (FQDN) pour l'hôte de destination** : CS-02.abdel.lan

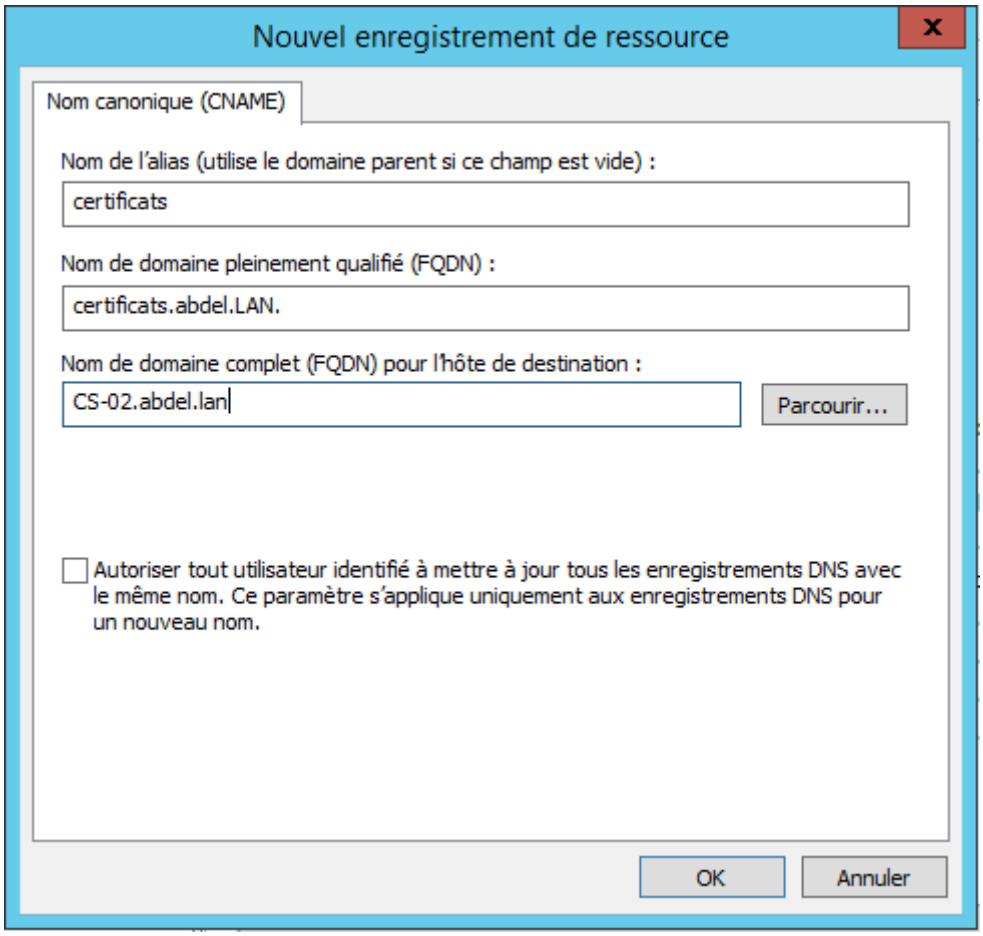


Figure IV.42. Nouvel enregistrement de ressources.

- **Étape 21** : Dans la console de gestion IIS, nous avons sélectionné le nom du serveur IIS, puis dans la partie centrale, nous avons fait un double clic sur **Authentification** afin d'activer l'**Authentification Windows**, et désactiver l'**Authentification anonyme** :

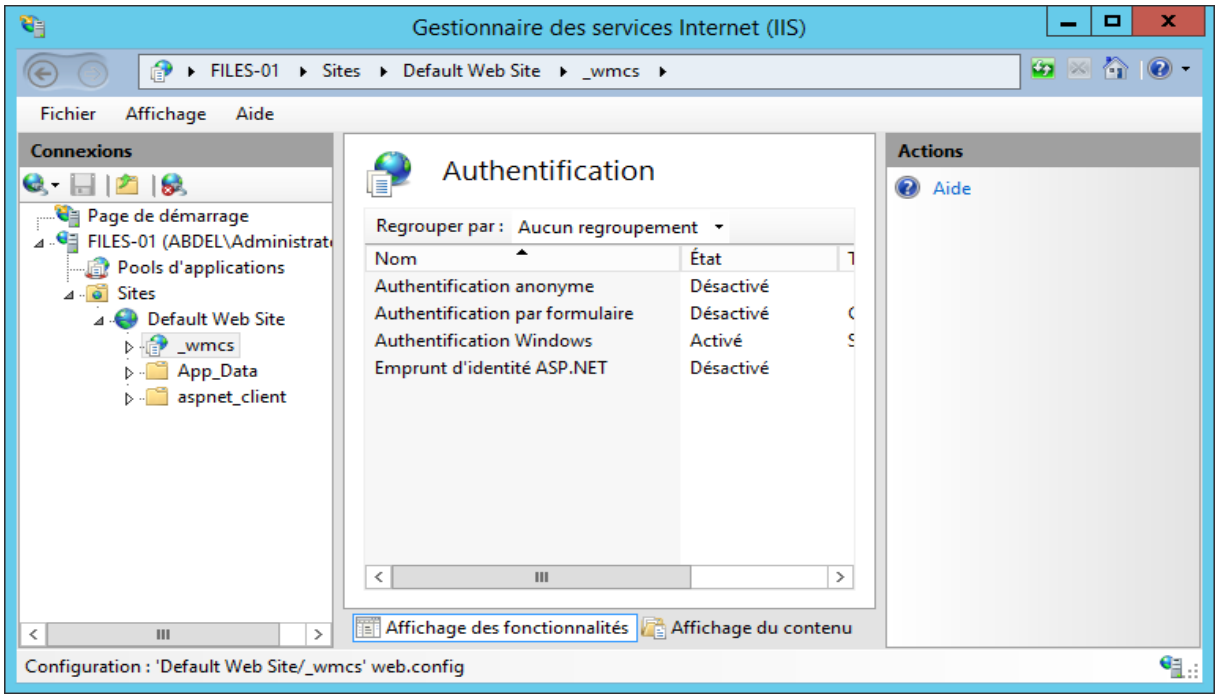


Figure IV.43. Authentification (IIS).

IV.9. Demander un certificat

Ces étapes nous permettent de demander un certificat auprès de l'autorité de certification émettrice. Le certificat à émettre doit être de type Basique EFS, pour l'utilisateur SERINE.

- **Étape 1** : Dans cette étape nous avons ouvert une session sur le serveur **CS-02** avec des identifiants d'administration du domaine *abdel.lan*. Puis nous avons démarré le Gestionnaire de serveur, ensuite cliqué sur **Outils** puis sur **Autorité de certification**.
- **Étape 2** : Dans cette étape nous avons développé l'arborescence de la console **Autorité de certification**, sélectionné le conteneur **Modèles de certificats**, ensuite nous avons fait un clic droit dessus puis cliquer sur **Gérer**.
- **Étape 3** : Dans la liste des modèles de certificats disponibles, nous avons fait un clic droit sur **EFS basique** et cliqué sur **Propriétés**.
- **Étape 4** : Dans la fenêtre **Propriétés de : EFS basique**, nous avons cliqué sur l'onglet **Sécurité**. Sélectionné le groupe **Utilisateurs authentifiés**, coché le droit **Inscrire** comme étant autorisé, puis cliqué sur **OK.**

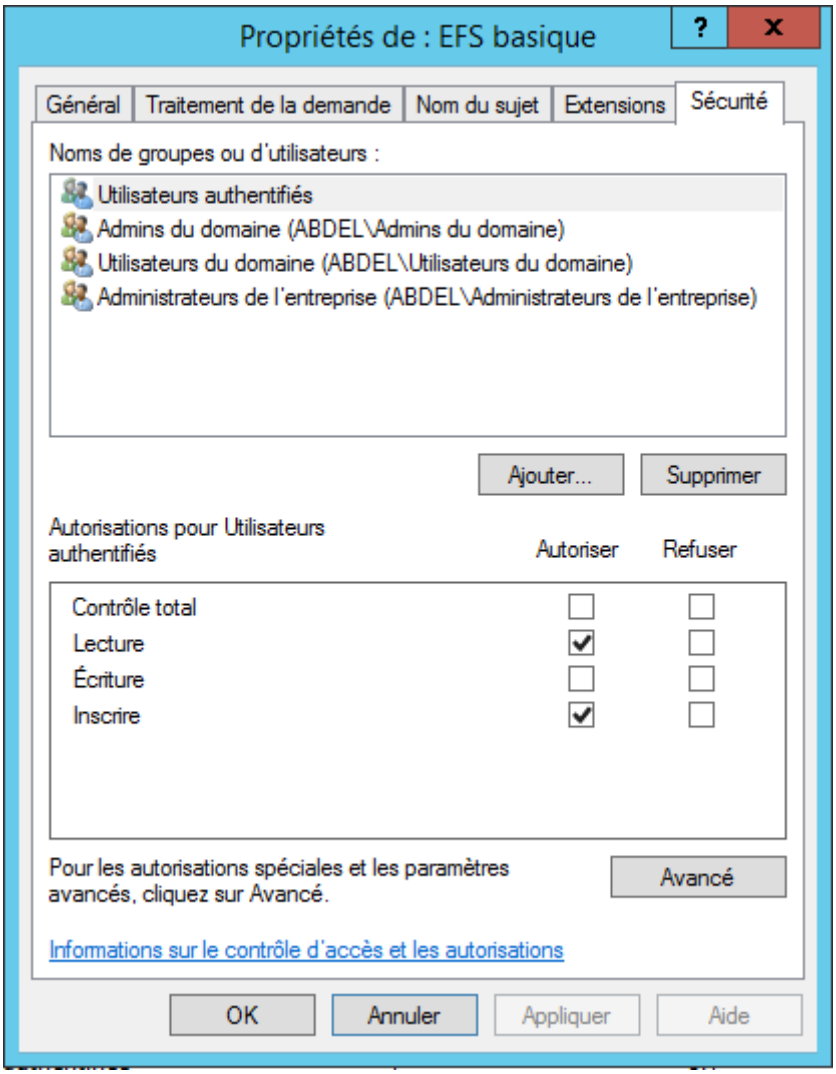


Figure IV.44. Propriété EFS basique.

- **Étape 5** : Dans cette étape nous avons fermé toutes les fenêtres de la console de gestion **Autorité de certification**.
- **Étape 6** : Dans cette étape nous avons ouvert une session sur le poste **CLIENT1** avec des identifiants d'utilisateur du domaine abdel.lan (se connecter avec le login : serine).
- **Étape 7** : dans la console MMC, nous avons ajouté le composant logiciel enfichable **Certificats**.
- **Étape 8** : Dans cette etape nous avons développé l'arborescence de la console afin de sélectionner le conteneur **Personnel**. Puis nous avons fait un clic droit dessus et cliqué sur **Toutes les tâches**, puis sur **Demander un nouveau certificat**.
- **Étape 9** : dans l'étape **Avant de commencer**, nous avons cliqué sur **Suivant**.
- **Étape 10** : dans l'étape **Sélectionnez la stratégie d'inscription de certificat**, nous avons sélectionné la stratégie Active Directory et cliqué sur **Suivant**.

- **Étape 11** : Dans cette étape nous avons coché la case correspondant au modèle de certificat EFS basique, puis cliqué sur **Inscription.**

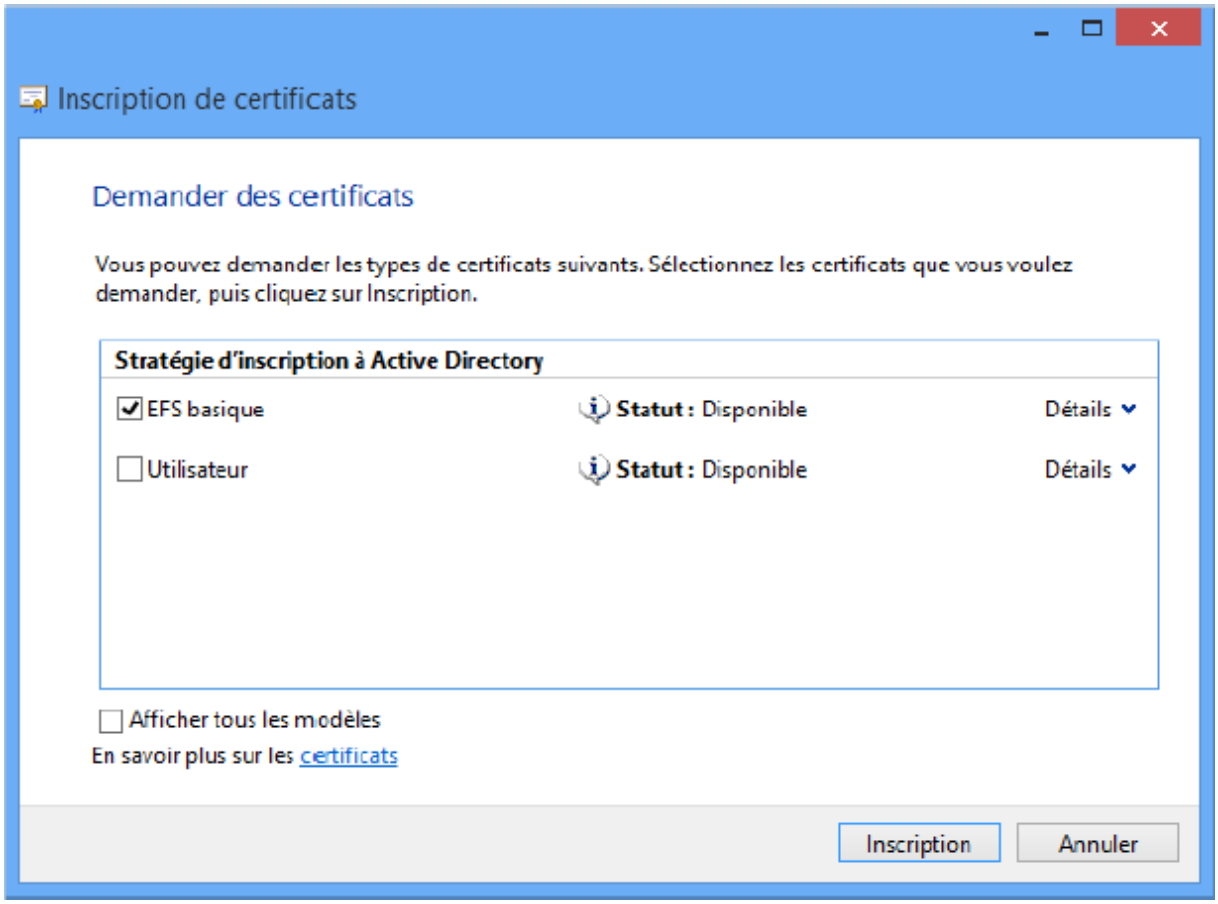


Figure IV.45. Inscription de certificats.

- **Étape 12** : Apres la réussite de l'opération, nous avons cliqué sur **Terminer**.
- **Étape 13** : Dans l'arborescence de la console Certificats, nous avons navigué dans le conteneur **Personnel/Certificats** afin d'apercevoir le certificat basique EFS délivré par l'autorité de certification *abdel-Emettrice-CA* pour l'utilisateur serine. Ce certificat peut désormais être utilisé pour chiffrer des données locales avec la technologie EFS (*Encrypting File System*) :

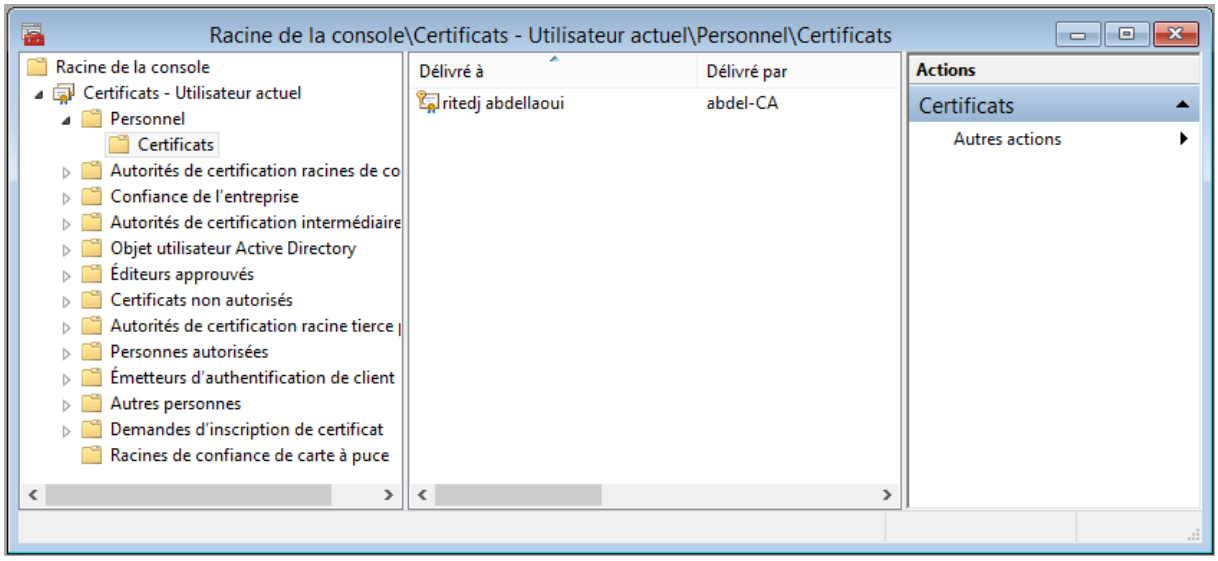


Figure IV.46. Certificat de chiffrement EFS.

IV.10. Discussion

L'objectif de ce travail été de faire une infrastructure d'une entreprise pour assurer le bon déroulement du travail de tous les utilisateurs et facilité le travail aux administrateurs système en toute sécurité en utilisant des clés et certificats, de ce fait, nous avons utilisé des outils de gérance des utilisateurs de façon qu'ils fassent leur travail convenablement et n'affectent pas le système.

Notre étude a permis de gérer les utilisateurs de système en toute sécurité, enfin un aspect pratique a été réalisé pour cette conception

Conclusion générale

Les réseaux informatiques sont devenus partie intégrante des différentes entreprises et deviennent un outil incontournable dans les sociétés modernes. Ils procurent un gain considérable du temps de même sur le plan financier. Utiliser un site de commerce électronique ouvre les portes et facilite la mise en place d'une activité internationale par la disponibilité et une circulation plus rapide de l'information.

Internet est le média d'aujourd'hui et de demain. Il est incontestablement devenu un formidable moyen de communication et de travail dans des conditions optimales. Il donne une nouvelle dimension aux entreprises. Il permet aux entreprises d'accroître leurs performances que ce soit en matière de productivité qu'en matière de rentabilité.
Mais, il se trouve que la mise au point d'un réseau informatique au sein d'une entreprise nécessite désormais une étude préalable approfondie. En effet, si la connexion est aujourd'hui simple et économique, la création d'un site demande des connaissances rigoureuses et permanentes du marché Internet et des besoins des internautes.
Il est également nécessaire de sécuriser la plate-forme contre le piratage informatique.

Notre travail nous a permis l'installation et la configuration d'un serveur de certificat, basé sur une infrastructure à clé publique PKI destiné à protéger l'accès au site web d'une entreprise via le port 443 et à chiffrer les données sur un disque dur.
Les utilisateurs peuvent consulter le site web en toute sécurité en https (protocole SSL), notamment pour transactions bancaires, le transfert de données et les informations de connexions, telles que les noms d'utilisateur et les mots de passe.
Toutefois, cette solution peut être étendue pour d'autres utilisation tel que :

- garantir l'identité d'un ordinateur à distance ;
- prouver votre identité à un ordinateur à distance ;
- garantir qu'un logiciel provient d'un éditeur de logiciel légitime ;
- empêcher les modifications d'un logiciel après sa publication ;
- sécuriser les e-mails ;
- signer des données et y ajouter un sceau d'horodatage ;
- sécuriser les communications sur Internet ;
- permettre toutes les politiques d'utilisation de clé
- permettre la signature OSCP.

Afin de renforcer d'avantage la sécurité du réseau de l'entreprise, il faut prévoir la mise en place et la configuration d'un firewall, système anti intrusion et l'accès à distance des utilisateurs grâce au VPN, car le protocole SSL ne peut prendre en charge à lui seul l'aspect de la sécurité.

[1] www.nicosphere.net/lunix

[2] http://lunixetleschoses.tuxfamily.org

[3] Benaissa M., 2011: thése architecture client-serveur,

[4]. http://www.guill.net

[5] Les bases de la Cryptologie Guenael Renault POLSYS LIP6/UPMC/INRIA 2016 page 03

[6]. Barsky D., 2006. Cryptographie générale.

[7]. Duquesne S., 2005. Cryptographie sur les courbes elliptiques.

[8]. http://math.univ-lyon1.fr/~roblot/masterpro.html /chapitre1. [En ligne] [Citation : 06 11 2012]

[9] : Ghislaine L., 2011. Introduction à la cryptologie.

[10] National Institute of Standards and Technology (NIST). FIPS PUB 46: The Data Encryption Standard, January 1977.

[11] National Institute of Standards and Technology (NIST). FIPS PUB 197: Advanced Encryption Standard, November 2001.

[12]. Cheikhrouhou O., Les Algorithmes Cryptographiques Symétriques, 2010-2011

[13]. Bayart F., La saga du DES. http://www.bibmath.net/crypto. [En ligne] [Citation : 09 03 1013]

[14]. SecuriteInfo.com, L'AES : Advanced Encryption Standard. http://www.securiteinfo.com/cryptographie/aes.shtml . [En ligne] [Citation : 11 03 2013]

[15]. Johnny .Lo, Li-Chang.,A framework for cryptography algorithms on mobile devices. s.l. : Faculty of Engineering, Build and Information Technology Department of Computer Science UNIVERSITY OF PRETORIA, 2007.

[16]. Emone J.-B., 2005. Algorithmes de chiffrement. 22 juin 2005.

[17] Diffie W. et M. Hellman. New Directions in Cryptography. IEEE Transaction on Information Theory, 22(6) :644–654, November 1976.

[18] Rivest R. L., A. Shamir, and L.M. Adleman., 1978. A Method for Obtaining Digital Signa- ture and Public-Key Cryptosystems. In Communications of the ACM, volume 21, pages 120–126.

[19] Diffie W. et M.E. Hellman. Multiuser Cryptographic Techniques. In AFIPS National Computer Conference, volume 45 of AFIPS Conference Proceedings, pages 109–112. AFIPS Press, 1976.

[20] Anssi A., Mécanismes cryptographiques – règles et recommandations concernant le choix et le dimmensionnement des m´ecanismes cryptographiques de niveau de robustesse standard, January 2010.

[21]. Nitaj A., 2009. CRYPTANALYSE DE RSA. Laboratoire de Mathématiques Nicolas Oresme, Université de Caen, France.

[22]. Asimane A., 2014. Windows Server 2012 R2 - Configuration des services avancés Editions ENI. 700 p.

Résumé :
Au cours de ce mémoire de fin d'étude, nous avons pu voir tous les aspects des services de certificats AD CS de
Microsoft Windows Server 2012 R2. Nous pourrons donc résumer de la manière suivante :
- Les autorités de certification permettent d'émettre et de gérer des certificats au travers du rôle de serveurAD CS.
- AD CS possède des rôles de services suivants :
- Autorité de certification
- Inscription de l'autorité de certification via le web
- AD CS peut être installé selon deux types d'autorités de certification :
- Autorité de certification autonome
- Autorité de certification d'entreprise
- Dans une hiérarchie d'autorité de certification à plusieurs couches, les autorités de certification de plus haut niveau doivent être mises hors ligne afin d'accroître la sécurité.
- Lorsqu'un certificat est présenté, le système vérifie d'abord sa validité en consultant la liste de révocation des certificats.

Mots clés : autorités de certification, IIS, ADCS, cryptographie, sécurité, plateforme WEB, certificats, architecture PKI.

Abstract :

Duringthisthesis, we were able to see all aspects of the AD CS Certificate Services fromMicrosoft Windows Server 2012 R2. So wecansummarize in the followingway:
-Certification authoritiesallow to issue and manage certificatesthrough the server roleAD CS.
-AD CS has the following service roles:
-Certification Authority
-Registration of the certification authority via the Web
-AD CS canbeinstalledaccording to two types of certification authorities:
-Autonomous Certification Authority
-Corporate Certification Authority
-In a multi-layered CA hierarchy, higher-level certification authorities must betaken offline to increasesecurity.
-When a certificateispresented, the system first checksitsvalidity by consulting the certificaterevocationlist.

Key words : certification authorities, IIS, ADCS, cryptography, security, WEB platform, certificates, Public key infrastructure.

(ALLOUACHE D.AMALOU A /ABDELLAOUI O.BOUSSADI A D) « 2018 »