وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR UNIVERSITY ANNABA

UNIVERSITE BADJI MOKHTAR ANNABA

جامعة باجي مختار – عنابة

Faculté des Sciences de l'Ingéniorat

Département d'Informatique

THÈSE

Présentée en vue de l'obtention du diplôme de
Doctorat 3ème cycle

Intitulé

Etude du protocole IPSec et métriques de sécurité

Filière: Informatique

Spécialité: Réseaux et Sécurité informatique

Par

Ahmim Marwa

DEVANT Le JURY

Directrice	Mme. Babes malika	Maitre de conférences à l'Université d'Annaba
Présidente	Mme. Djellab Natalia	Professeur à l'Université d'Annaba
Examinateur	Mme. Merouani Hayet Farida	Professeur à l'Université d'Annaba
Examinateur	Mr. Derdour Makhlouf	Maitre de conférences à l'Université de Tébessa
Examinateur	Mr. Benmohammed Mohammed	Professeur à l'Université de Constantine

Année 2016

Dédicace

Je dédie ce travail

- *A mon grand père;*
- *A mes très chers parents que j'aime plus que tout au monde.*

Remerciement

Je remercie d'abord la grâce de dieu, le clément et Miséricordieux pour m'avoir guidé, éclairé sur la bonne voie du savoir et de la recherche scientifique pour pouvoir mener à terme ce travail modeste.

Je tiens en premier lieu à exprimer toute ma gratitude à mon Encadreur Madame Babes Malika, maitre de conférences à l'Université d'Annaba pour son soutien et ses bons conseils sans lesquels il m'aurait était impossible d'achever ce projet de recherche.

Je remercie vivement Madame Ghoualmi Nacira, professeur à l'Université d'Annaba, pour sa disponibilité indéfectible, sa gentillesse, ses aides ainsi que ses remarques pertinentes et instructives.

Je souhaite exprimer ma gratitude envers Madame Djellab Natalia, professeur à l'Université d'Annaba, pour m'avoir fait l'honneur d'être la présidente du jury.

Remerciement

Je remercie chaleureusement Madame Merouani Hayet Farida, professeur à l'Université d'Annaba, Monsieur Makhlouf Derdour, maitre de conférences à l'Université de Tebessa, Monsieur Benmohammed Mohammed, professeur à l'Université de Constantine 2, de m'avoir fait l'honneur d'accepter d'examiner ma thèse.

Ma gratitude et mes vifs remerciements à ma mère, mon père ainsi que toute ma famille pour m'avoir encouragé durant toute la durée de mes études et pour m'avoir apporté un soutien indispensable à ma réussite.

Je remercie du plus profond de mon cœur tous mes amis pour leur grand soutien.

الملخص

موضوع أطروحتنا هام جدا في مجال امن الحاسوب, إذ يشمل دراسة IPSec و المقاييس الأمنية, و تم تعريف IPSec من قبل IETF (Internet Engineering Task Force) منذ عام1992.في عام 1995، تم تطوير أول إصدار أساسي علي شكل RFC (Request For Comment) دون جزء إدارة المفاتيح حيث تم اكتشاف العديد من الإخفاقات في هذا الإصدار, في عام 1998 تم إدخال أول تحسين لمعالجة عيوب هذا البروتوكول ، حيث تم اقتراح بروتوكول IKEv1، الذي يضيف نظام ديناميكي لإدارة الإعدادات السرية ل IPSec.

في هذه الأطروحة اهتممنا في البداية بالتهيئة الديناميكية IPSec حيث درسنا مختلف البروتوكولات الموجودة في الأدبيات. و قد قدمنا طريقة تشغيل كل بروتوكول تليها دراسة مقارنة البروتوكولات من حيث السلامة والفعالية، لتقييم السلامة استخدمنا كمقياس أمن عدد من الهجمات على متطلبات أمن بروتوكول إدارة المفاتيح، و لتقييم الأداء استخدمنا كمقياس أداء عدد تبادل الرسائل والعمليات المستخدمة لتبادل. في الاقتراح الأول ل IKE، إذ ركزنا جهودنا على ثلاثة أنواع من الهجمات: DOS, Main-in-the-Middle و Replay. هذا الاقتراح قد قاوم هذه الهجمات، ولكنه لا يفي بعدد من متطلبات السلامة PFS، CK و KCI، علاوة على تعقيدات الكبيرة للبروتوكول.

وقد مكننا هذا التوليف بلقتراح بروتوكول IKE جديد قائم على التشفير باستخدام المنحنى البيضاوي، حيث انه يستوفي عدة متطلبات الأمن ويقاوم أنواع مختلفة من الهجمات مثل DOS, Main-in-the-Middle, Replay, Modification وReflection. بالإضافة لهذا فإنه ضمن مستوى عال من الأمن مع مفتاح صغير وأقل حساب ويستخدم ثلاث رسائل لإنشاء رابطة الأمن IKE ورسالتين لتأسيس رابطة الأمن IPSec. لجعل IKE أخف وفعال جدا في شبكات الاستشعار، اقترحنا مخطط آخر ل IKE المسمي EIAKEP. ثم اقترحنا إتباع نهج جديد كإجراء أمني على أساس التصنيف والتحلل لسياسات الأمنية.

كلمات البحث: أمن الحاسوب، IPSec، مقاييس الأمن، متطلبات السلامة, تدابير السلامة ,هجوم الكمبيوتر IKE,.

Abstract

The subject of our thesis deals with a very important center of research in the field of computer security, which combines the study of IPSec protocol and security metrics. IPSec has been defined by the IETF (Internet Engineering Task Force) since 1992. In 1995, a first basic version was developed as an RFC (Request For Comment) without the key management part. Many failures are discovered in this version. To deal with these shortcomings a first improvement of this protocol was introduced in 1998, where they have proposed the IKEv1 protocol, which allows adding a dynamic system for the IPSec privacy settings management.

In this thesis; we are interested, in a first time, to the dynamic initialization phase of IPSec protocol, where we have examined the various existing protocols in literature. Then, we have presented the operation mode of each protocol followed by a comparative study of the protocols in terms of safety and efficiency. To evaluate the safety, we have used as security metric, the number of attacks added to the security requirements of the key management protocol. And to evaluate performances, we have used as efficiency metric, the number of message exchange added to the operations used to ensure exchange.

In the first proposal of the IKE protocol, we have focused our efforts on three types of attacks: DoS, man in the middle and replay. This proposal may resist to these attacks, but it does not meet several security requirements (PFS, CK, and KCI); furthermore, the complexity of the protocol is high. This synthesis allowed us to propose a new IKE protocol based on elliptic curve cryptography, this last one satisfies several security requirements and resists to various attack types such as modification, reflection, replay, DoS and man-in-the-middle.

Also, it ensures a high level of security with smaller key size and lower computation cost. It uses three messages to create the SA IKE and two messages to establish the SA IPSec. In order to make the IKE lighter and very effective in sensor networks, we

Abstract

have proposed another IKE protocol schema called EIAKEP (An efficient Internet authenticated-key exchange protocol).

Then, we have proposed a new approach of security measurement based on classification and decomposition of security policy.

Keywords: Computer Security, IPSec, Security metrics, Computer attack, Security measure, Security requirement, IKE.

Résumé

Le sujet de notre thèse traite un axe de recherche très important de la sécurité informatique qui combine l'étude du protocole IPSec et les métriques de sécurité. L'IPSec a été défini par le groupe de travail d'IETF (Internet Engineering Task Force) depuis 1992. En 1995, une première version basique a été développée sous forme de RFC (Request For Comment) sans la partie de la gestion des clés. Plusieurs défaillances sont découvertes dans cette version. Afin de remédier à ces défauts une première amélioration de ce protocole a été introduite en 1998, où ils ont proposé le protocole IKEv1, qui permet d'ajouter un système dynamique pour la gestion des paramètres de confidentialité de l'IPSec.

Dans cette thèse, nous nous sommes intéressés en premier temps à la phase d'initialisation dynamique du protocole IPSec où nous avons examiné les différents protocoles existants dans la littérature. Nous avons présenté le fonctionnement de chaque protocole suivi par une étude comparative de ces protocoles en termes de sécurité et d'efficacité. Pour évaluer la sécurité nous avons utilisé comme métrique de sécurité le nombre d'attaques plus les exigences de sécurité du protocole de la gestion des clés. Pour évaluer les performances nous avons utilisé comme métrique d'efficacité le nombre d'échange des messages plus les opérations utilisées pour assurer l'échange.

Dans la première proposition du protocole IKE, nous avons focalisé nos efforts sur trois types d'attaque: DoS, l'homme du milieu et par rejeu. Cette proposition peut résister à ces attaques, mais elle ne satisfait pas plusieurs exigences de sécurité (PFS, CK et KCI), en plus la complexité du protocole est élevée. Cette synthèse, nous a permis de proposer un nouveau protocole IKE basé sur la cryptographie à courbe elliptique, ce dernier satisfait plusieurs exigences de sécurité (PFS, CK et KCI) et résiste à plusieurs types d'attaques telles que : DoS, l'homme du milieu, par rejeu, la modification et par réflexion. De plus, il assure un niveau de sécurité élevé avec une

Résumé

clé de petite taille et moins de calcul modulaire. Il utilise trois messages pour créer l'association de sécurité IKE et deux messages pour établir l'association de sécurité IPSec. Afin de rendre le protocole IKE plus léger et très efficace dans les réseaux de capteurs, nous avons proposé un autre schéma du protocole IKE nommé EIAKEP.

Ensuite, nous avons proposé une nouvelle approche de mesure de sécurité basée sur la classification et la décomposition des politiques de sécurité.

Mots clés: Sécurité informatique, IPSec, IKE, Métrique de sécurité, Mesure de sécurité, Exigence de sécurité, Attaque informatique.

Table des matières

Liste des figures

Liste des figures

Liste des figures

Liste des tableaux

Introduction générale

1. Le cadre scientifiques

Avec le développement rapide de l'Internet, la sécurité de l'information et la protection de la vie privée sont devenues un problème crucial pour les organisations et les entreprises. Différents protocoles et outils de sécurité ont été développés et appliqués par plusieurs organisations afin de protéger leurs données confidentielles, services et ressources contre les attaques passives et actives. Parmi ces protocoles on recense: le protocole TLS, SSH et IPSec.

Dans notre thèse, nous nous intéressons au protocole IPSec. L'IPSec est l'un des protocoles de l'IETF qui permettent d'assurer des communications sécurisées dans des réseaux non sécurisés tels que LAN, WAN. L'IPSec est le seul protocole qui peut être déployé sans aucune modification des réseaux existants. Ce protocole est conçu pour assurer l'authentification de la source de données, l'intégrité et la confidentialité des données transmises dans les réseaux, et la protection des réseaux contre le rejeu.

L'IPSec peut fonctionner en deux modes: transport et tunnel. Le mode transport est utilisé pour les communications de bout en bout (Host to Host). Le mode tunnel est utilisé dans les configurations passerelle à passerelle ou passerelle à hôte (Gate-To-Gate ou Host-To-Gate).

L'IPSec fournit ces services de sécurité à travers deux extensions d'en-têtes: l'Authentication Header (AH) (Kent et Atkinson, 1998a) et l'encapsulating security payload (ESP) (Kent et Atkinson, 1998b).

Avant de fournir les dimensions de la sécurité, les deux entités de l'IPSec doivent se mettre d'accord sur la nature de la sécurité à appliquer, telle que: les algorithmes cryptographiques à utiliser, les en-têtes de sécurité (AH, ESP, ou les deux) à appliquer, les clés secrètes, et le mode transport ou tunnel. Pour gérer ces paramètres confidentiels, l'IPSec utilise le concept de l'association de sécurité (Security Association, SA). Une SA est une relation à sens unique entre l'émetteur et le récepteur. Elle contient toutes les informations nécessaires pour le traitement d'IPSec sur un paquet IP. Donc, l'IPSec a besoin d'un moyen d'échanger des informations de l'association de sécurité. Le protocole Internet Key exchange (IKE)

est utilisé par l'IPSec afin de lui fournir ces capacités de manière dynamique et sécurisée.

Le protocole IKE est le cœur de l'architecture de l'IPSec. Il est utilisé pour créer des associations de sécurité (SA) qui définissent comment le trafic entre les deux parties doit être protégé. L'exigence principale de sécurité pour le protocole IPSec dépend du protocole IKE.

2. La problématique

Les paquets circulent de façon publique et bien connue sur le réseau Internet, par conséquent, n'importe quel paquet peut être capturé par un hacker. Donc, il est relativement facile de falsifier les adresses des paquets IP, modifier le contenu des paquets IP, rejouer un vieux paquet IP, et inspecter le contenu des paquets IP en transit. Afin de pallier à ces faiblesses, plusieurs protocoles de sécurité ont été proposés, parmi ces protocoles, nous intéresserons au protocole IPSec.

L'IPSec a été défini par le groupe de travail d'IETF (Internet Engineering Task Force) depuis 1992. En 1995, une première version basique a été développée sous forme de RFC (Request For Comment) sans la partie de la gestion des clés. Plusieurs défauts ont été déclarés dans cette version; afin de les corriger, une première amélioration de ce protocole a été faite en 1998, où ils ont proposé le protocole IKEv1, qui permet d'ajouter un système dynamique pour la gestion des paramètres confidentiels de l'IPSec. Malheureusement, la première version du protocole IKEv1 a été critiquée par plusieurs chercheurs, en particulier la vulnérabilité aux attaques passives et actives.

Dans le but d'améliorer la sécurité de l'IKEv1, plusieurs protocoles IKE ont été proposés. Malgré la richesse de la littérature de ces protocoles IKE amélioré ces protocoles souffrent encore de certaines faiblesses. Afin de palier ces faiblesses, nous proposons trois protocoles basés sur la métrique de sécurité «Nombre d'attaques». De plus, nous avons proposé une nouvelle approche basée sur la classification et la décomposition des politiques de sécurité.

3. Le but du travail et le travail réalisé

L'objectif de notre travail est d'étudier le protocole IPSec et les métriques de sécurité afin de proposer une amélioration de celui-ci. Nous avons concentré notre travail sur la phase d'initialisation du protocole IPSec qui représente le cœur de son architecture, cette phase se base sur le protocole IKE.

Nous avons fait une analyse théorique et formelle sur l'original IKE ainsi que sur ces successeurs. Cette analyse se base sur la métrique de sécurité « Nombre d'attaques ». Ensuite, nous avons proposé trois contributions : dans la première proposition, nous avons concentré nos efforts sur les 3 types d'attaques: l'attaque par rejeu, l'attaque DoS et l'attaque de l'homme du milieu (man-in-the-middle). Afin de rendre le protocole IKE plus efficace et plus sécurisé, nous avons focalisé nos recherches pour la deuxième et la troisième propositions sur la complexité du protocole et le nombre d'échanges ainsi que les 5 types d'attaques qui sont les suivantes: la modification, par réflexion, par rejeu, DoS et l'homme du milieu (man-in-the-middle). Enfin, nous avons proposé une approche pour évaluer la sécurité de l'IPSec basé sur la classification et la décomposition hiérarchique de la politique de sécurité.

4. La structure de la thèse

Notre thèse est composée de cinq chapitres. Dans le premier chapitre, nous introduisons les concepts élémentaires de la sécurité des réseaux informatiques où nous présentons les dimensions de la sécurité, les différents types d'attaques ainsi que les différents mécanismes cryptographiques.

Après avoir introduit les éléments fondamentaux et les principes de base de la cryptographie, nous présentons dans le deuxième chapitre les différentes technologies et les protocoles qui peuvent être utilisés pour fournir des services de sécurité pour les réseaux IP où nous nous focalisons sur les protocoles qui nous permettent de sécuriser les échanges au niveau de la couche réseau tel que le protocole IPSec.

Les propriétés de sécurité de l'IPSec dépendent essentiellement des protocoles d'échange des clés sous-jacents connus sous le nom du protocole IKE. Dans le troisième chapitre, nous donnons une description détaillée de ce protocole, puis nous

examinons ses successeurs [IKEv1, JKF, IKE proposé par Haddad et al. en 2004, IKEv2 et le protocole IKE proposé par Ray et al. en 2012] afin de les comparer à nos contributions proposées dans le but de renforcer la sécurité de l'IPSec.

Lord Kelvin a dit « *si vous ne pouvez pas mesurer, vous ne pouvez pas améliorer* », donc pour obtenir des preuves sur le niveau de sécurité d'un système ou des services une approche de mesure de sécurité est nécessaire. Dans le quatrième chapitre, nous présentons les différents concepts de la métrique de sécurité tels que : les différentes définitions des métriques et des mesures existantes dans la littérature, les propriétés des métriques de sécurité, les objectifs de la mesure de sécurité et les dimensions mesurables. De plus, nous montrons une étude des différentes approches de mesure de sécurité proposées dans la littérature afin de proposer une approche qui permet de mesurer la sécurité de l'IPSec. Enfin, nous présentons notre contribution des métriques d'authentification qui se base sur le travail de (Wang and Wulf, 1996) et (Savola and Abie, 2009). Cette contribution a été présentée dans la conférence l'Optimisation et les Systèmes d'Information, Béjaia, Algérie (2014).

Dans le cinquième chapitre, nous présentons nos trois contributions du protocole IKE qui permettent d'améliorer la sécurité du protocole IPSec ainsi qu'une nouvelle approche d'évaluation de la sécurité basée sur la politique de sécurité.

Chapitre I : La sécurité des réseaux informatique

Dans ce premier chapitre, nous introduisons les concepts élémentaires de la sécurité des réseaux informatiques. Nous commençons par décrire les dimensions de sécurité (l'authentification, la confidentialité des données, l'intégrité des données, la non-répudiation, le contrôle d'accès et la disponibilité). Ensuite, nous présentons les différents types de vulnérabilité et d'attaque. Après avoir discuté les ennuis causés par les attaques, nous détaillons les différents mécanismes cryptographiques.

1.1. Propriétés de la sécurité

L'UIT (Union internationale des télécommunications) a défini un ensemble large de dimension de sécurité qui comprend: l'authentification, la confidentialité des données, l'intégrité des données, la non-répudiation, le contrôle d'accès et la disponibilité (ITU-T, 2003).

- **L'authentification**: elle est le processus de sécurité qui vérifie l'identité d'une entité demandant l'accès aux ressources ou applications du système. L'entité peut être une personne ou non-personne comme routeur, pare-feu, application, bases de données ou un autre composant du système. Il existe de nombreux mécanismes ou facteurs d'authentification:
  - **Ce que je sais**: par exemple, un mot de passe;
  - **Ce que je sais faire**: une signature manuscrite sur écran tactile/digital;
  - **Ce que je suis**: une caractéristique physique comme une empreinte digitale;
  - **Ce que je possède**: une carte à puce par exemple.

- **La confidentialité**: la confidentialité est une propriété de sécurité qui stipule qu'une information ne peut être utilisée ou lue que par sa destination et non par une autre entité. Assurer la confidentialité des données implique l'implémentation des techniques empêchant toute délibération ou divulgation de l'information à des utilisateurs non autorisés. Ces derniers sont appelés les mécanismes de chiffrement qui sont détaillés dans la section (1.3).

- **L'intégrité**: l'intégrité est un service de sécurité qui assure que les informations transmises entre la source et la destination n'ont pas été altérées. Garantir

l'intégrité des données implique la prévention et la détection de la modification, l'ajout ou la suppression des informations.

- **La non-répudiation**: c'est l'ensemble des contrôles nécessaires pour prévenir la répudiation. Typiquement, la non-répudiation se réfère à la capacité d'empêcher une personne ou une entité ayant effectué des opérations sur des données de pouvoir nier l'avoir fait.

- **Le contrôle d'accès**: c'est l'ensemble des politiques de sécurité nécessaires pour prévenir l'utilisation non autorisée d'une ressource du réseau. Le contrôle d'accès consiste à assurer que seules les personnes ou les entités autorisées peuvent accéder aux éléments du réseau, aux services, aux flux d'informations et aux applications.

- **La disponibilité**: la disponibilité est la capacité d'assurer que les utilisateurs ont un accès rapide et continu aux éléments du réseau, aux informations, aux services et aux applications. Garantir la disponibilité des systèmes, implique l'implémentation des techniques empêchant toute tentative de déni de service.

1.2. Vulnérabilité, Attaques

Une vulnérabilité est une faiblesse identifiée dans un système contrôlé, où les mécanismes de contrôle ne sont pas présents ou ne sont plus efficaces (ISO/IEC 27000, 2009).

Une attaque est un acte qui exploite une faille pour compromettre un système contrôlé. Il est réalisé par un attaquant qui détruit, modifie, expose, vole ou obtient un accès non autorisé aux informations, aux services, aux flux d'informations et aux applications d'une organisation. Comme l'illustre la Figure 1.1; on distingue deux types d'attaques, et chacune d'entre elles se subdivise en plusieurs catégories:

- **L'attaque passive**: ce type d'attaque consiste à analyser et surveiller les communications non protégées, décrypter le trafic au cryptage faible et capturer les informations d'authentification (comme les mots de passe). L'attaque peut entraîner la divulgation des informations à un attaquant sans l'autorisation ou la

connaissance de l'utilisateur. Les exemples des attaques passives incluent la divulgation de renseignements personnels tels que des numéros de carte de crédit, mot de passe des E-mails et les dossiers médicaux (Van Quang, 2005; Cole et al., 2005).

- **L'attaque active**: ce type d'attaque a pour but de contourner ou de briser les fonctions de protection, voler ou modifier les informations, et introduire un code malveillant. L'attaque active peut entraîner la divulgation ou la diffusion des fichiers de données, le déni de service, ou la modification de données (Van Quang, 2005; Cole et al., 2005).

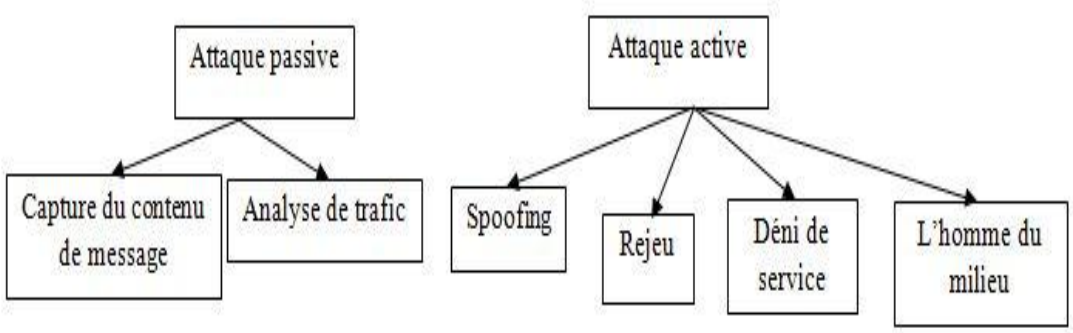

Figure 1.1 Différents types d'attaques (Van Quang, 2005)

Les instanciations spécifiques de ces types d'attaques sont:

- Le déni de service (DoS): l'attaquant envoie un grand nombre de connexions ou des demandes d'informations, afin de rendre la machine cible surchargée pour qu'elle ne puisse pas répondre aux demandes légitimes de service (voir Figure 1.2). La machine cible peut se bloquer ou simplement devenir incapable de remplir les fonctions ordinaires (Michael and Herbert, 2009).

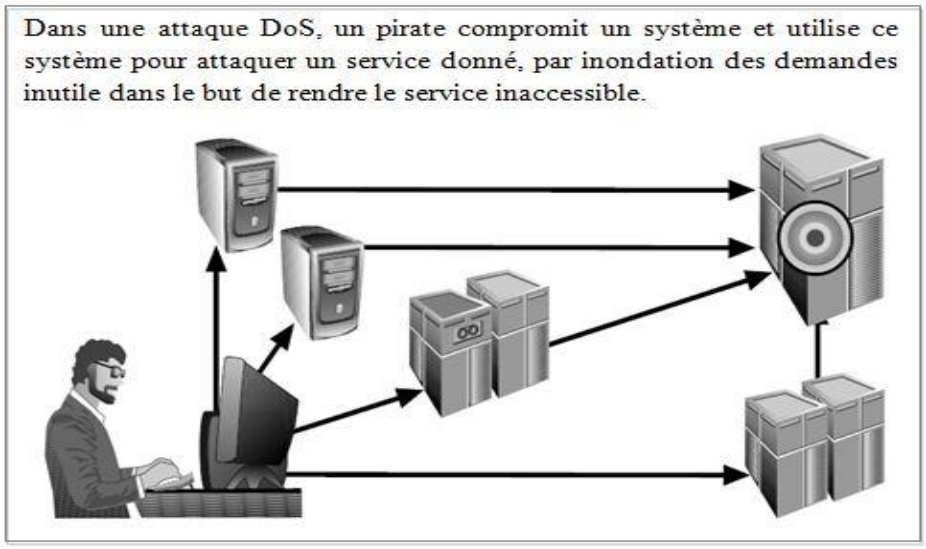

Figure 1.2 Attaque de déni de service (Michael and Herbert, 2009)

- L'attaque de l'homme du milieu (HDM) ou man-in-the-middle (MITM): se produit quand un attaquant (Carol) se positionne entre les deux entités de communications (Alice et Bob) dans le but d'espionner, modifier, supprimer, re-router, ajouter, forger, ou détourner des données échangées entre eux (Michael and Herbert, 2009).

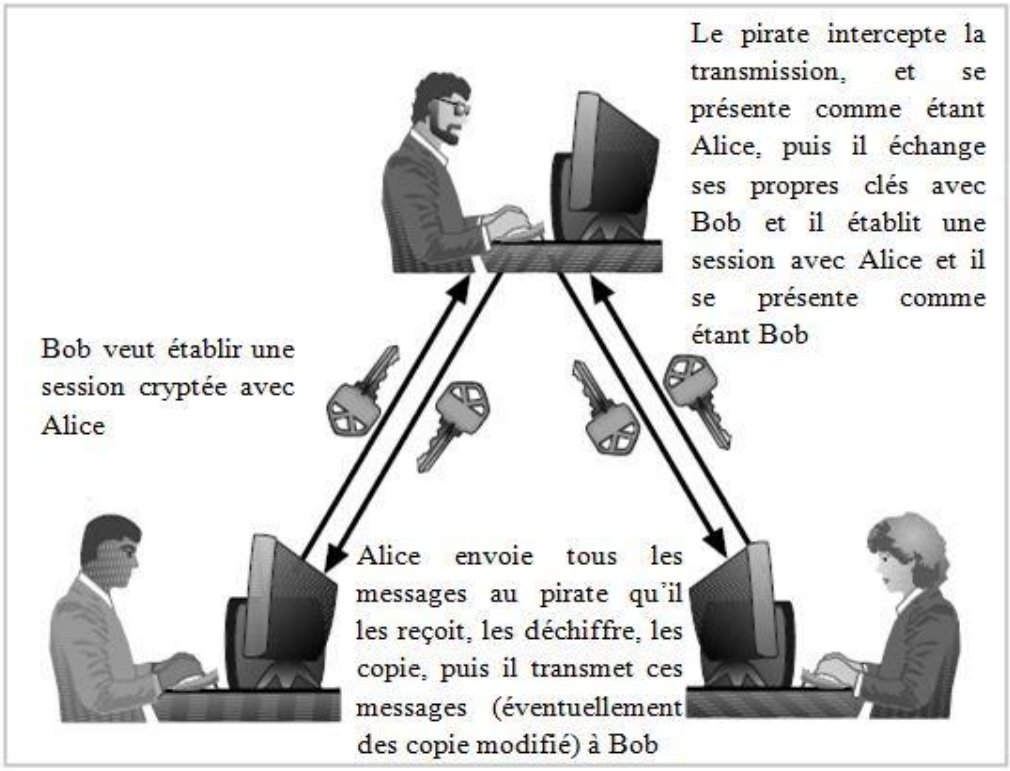

Figure 1.3 Attaque de l'homme du milieu (Michael and Herbert, 2009)

- L'attaque par rejeu ou replay attack : l'attaquant intercepte des paquets de données et les rejoue dans le but d'usurper l'identité ou les droits d'un utilisateur légitime. Un attaquant (Carol) met sur écoute les échanges transmis entre la source (Alice) et la destination (Bob), puis il utilise les messages enregistrés pour communiquer avec Alice et Bob (Cole et al., 2005).
- L'attaque par réflexion: cette attaque consiste à utiliser l'IP spoofing et l'ICMP afin de saturer un réseau cible avec le trafic, en lançant une attaque DoS. Elle se compose de trois éléments : le site source, le site de rebond, et le site cible. L'attaquant (le site source) envoie un paquet de ping usurpé à l'adresse de diffusion d'un grand réseau (le site de rebond). Ce paquet modifié contient l'adresse du site cible. Cela provoque le site de rebond pour diffuser les fausses informations pour tous les appareils sur son réseau local. Donc tous ces dispositifs répondent avec une réponse de type « replay » au système cible, qui sera saturé (Cole et al., 2005).

1.3. Mécanisme de sécurité cryptographique

La cryptographie est un ensemble de techniques permettant d'assurer les dimensions de sécurité par l'utilisation des méthodes de chiffrement. Le chiffrement permet de transformer le texte clair en un texte chiffré à l'aide d'un algorithme de chiffrement et d'une clé (voir Figure 1.4). Le déchiffrement permet de transformer le texte chiffré en texte clair identique à celui d'origine à l'aide d'un algorithme de déchiffrement et une clé (Baudet, 2007 ; Lafourcade, 2006).

Figure 1.4 Processus général de cryptage

Les algorithmes de chiffrement et de déchiffrement sont basés sur des problèmes mathématiques difficiles à résoudre. De plus, il est difficile de déchiffrer le message chiffré sans connaitre la clé secrète. En général, les méthodes cryptographiques sont divisées en deux grandes catégories: la cryptographie symétrique dite à clé secrète et

la cryptographie asymétrique dite à clé publique. De nos jours, les systèmes cryptographies utilisent une combinaison hybride des algorithmes symétriques et asymétriques.

1.3.1. Chiffrement symétrique

Dans cette méthode de chiffrement, la clé utilisée pour le chiffrement et le déchiffrement du message est la même. Le schéma suivant illustre le fonctionnement de la méthode du chiffrement symétrique (Baudet, 2007; Lafourcade, 2006).

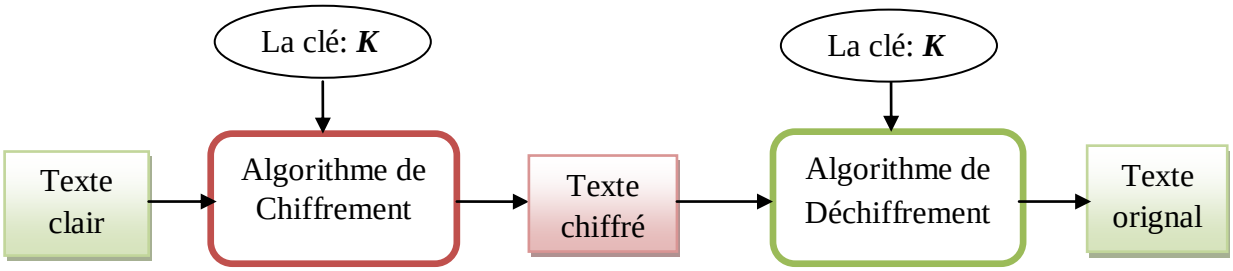

Figure 1.5 Chiffrement symétrique

Les méthodes de chiffrement symétrique utilisent des opérateurs mathématiques tells que la substitution et la transposition qui peuvent être programmées dans des algorithmes de calcul extrêmement rapides afin que le processus de chiffrement et de déchiffrement soit exécuté rapidement même dans de petits ordinateurs. L'inconvénient majeur de cette méthode de chiffrement est qu'il faut partager une clé secrète entre l'initiateur et le répondeur en toute sécurité. Cependant, si l'une des copies de la clé tombe dans de mauvaises mains, les messages peuvent être déchiffrés par d'autres personnes et l'initiateur et le répondeur prévus ne vont pas être au courant que le message a été intercepté. Il existe plusieurs algorithmes populaires de chiffrement symétriques. Parmi ces algorithmes, nous présentons DES (Data Encryption Standard), 3DES (Triple DES) et AES (Advanced Encryption Standard).

1.3.1.1. Le Data Ecryption Standard

Le Data Ecryption Standard (DES) représente l'algorithme de cryptage symétrique le plus employé, ces dernières années, par les différents systèmes et protocoles de sécurité. En 1973, le NBS (National Bureau of Standards) a lancé un appel d'offres pour développer un algorithme standard de chiffrement afin de sécuriser les communications sensibles. La société IBM (International Business Machines) a

proposé une famille d'algorithmes de chiffrement appelée Lucifer développé par Horst Feistel. L'un de ces algorithmes suggéré pour cette compétition d'appel d'offres est retenu pour devenir le DES. Une amélioration de l'algorithme Lucifer d'IBM est conçue par NSA (National Security Agency), précisément en réduisant la taille de la clé à 56 symboles binaires au lieu des 112 qui ont été proposé par IBM.

DES est un chiffrement par bloc qui est conçu pour crypter et décrypter des blocs de données composés de 64 bits, en utilisant une clé de 56 bits. Le processus de chiffrement est constitué de deux permutations, que nous appelons permutations initiale et finale, et 16 tours de Feistel. Chaque tour utilise une clé de taille 48 bits et des opérations de transposition, de substitution et de chiffrement de Vernam. La figure 1.6 montre les différents éléments du chiffrement DES. L'EFF (Electronic Frontier Foundation) à découvrir la vulnérabilité du chiffrement DES a une attaque par force brute (recherche de clé) en utilisant une machine conçue à ce but. Pour résoudre cette vulnérabilité, une amélioration de DES a été proposée (Guillot, 2013; Lafourcade, 2006; Thuillet, 2012).

1.3.1.2. Le triple Data Ecryption Standard

Data Encryption Standard (DES) utilise une clé de taille 56 bits, considérée, malheureusement insuffisante pour chiffrer les données sensibles en raison des progrès des moyens de calculs. Avec des machines performantes et en temps raisonnable, l'attaque par force brutale rend la clé DES envisageable. Pour pallier cette lacune, la NSA a proposé un algorithme sans concevoir un nouveau système de cryptage, c'est le triple DES (Guillot, 2013; Lafourcade, 2006; Thuillet, 2012).

3 DES étend simplement la taille de clé du DES en appliquant l'algorithme DES trois fois de suite avec trois clés différentes. La taille de la clé combinée est égale à 168 bits (trois fois 56).

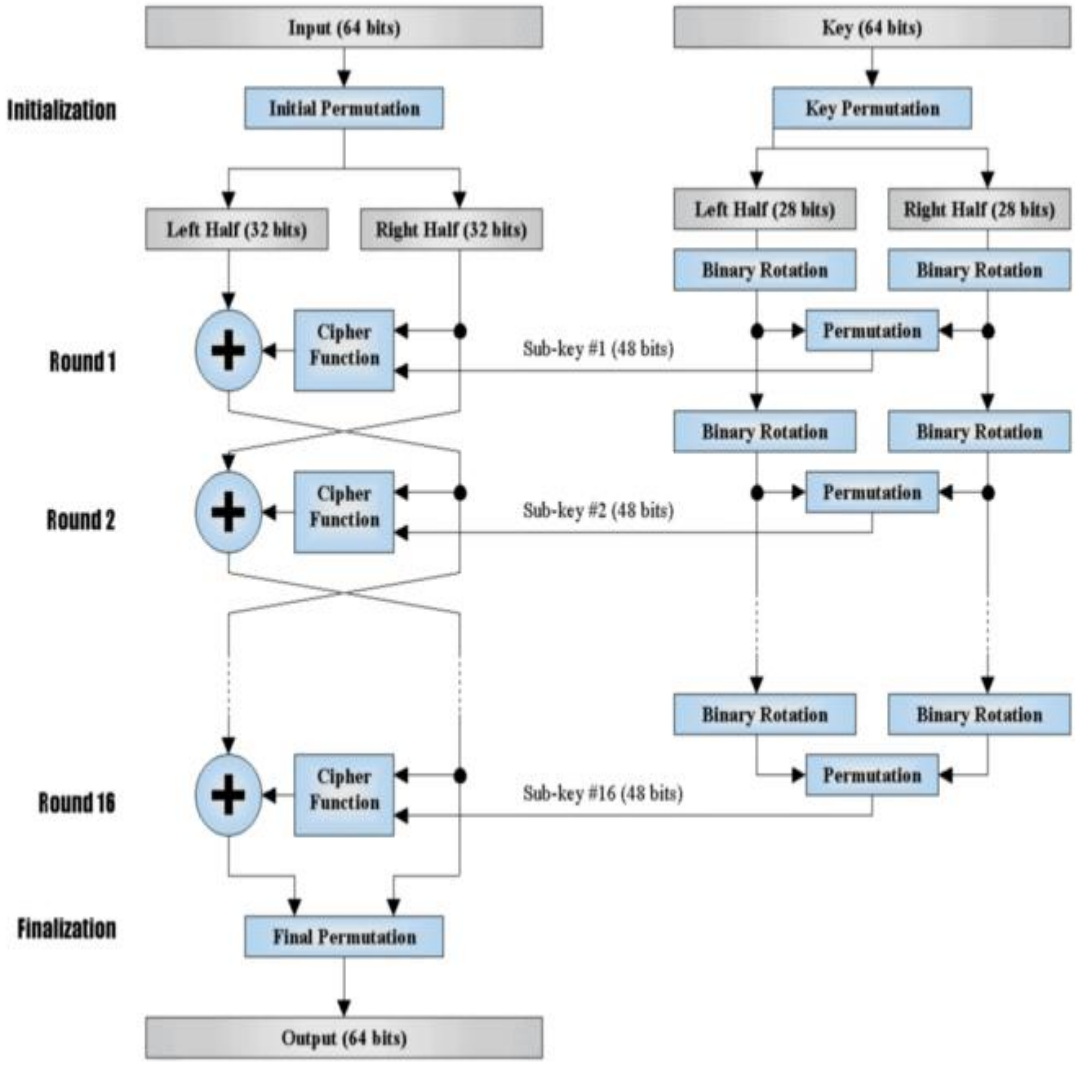

Figure 1.6 DES: schéma de fonctionnement (Guillot, 2013)

1.3.1.3. L'Advanced Encryption Standard (AES)

De nos jours, l'AES est l'algorithme le plus utilisé dans de nombreux standards de sécurité comme SSH, IPsec. En 1997, le NIST (National Institute of Standards and Technology) a mené un concours visant à développer un remplaçant de DES. L'objectif de cet appel consiste à fournir un nouvel algorithme de chiffrement plus sûr et efficace utilisable dans le monde civil et commercial. Après l'évaluation de 15 candidats par NIST, le candidat Rijndael conçu par les cryptologues Belges Vincent Rijmen et Joan Daemen est choisi pour être le nouveau Advanced Encryption Standard en 2000 (Guillot, 2013; Lafourcade, 2006; Thuillet, 2012).

AES est un algorithme de chiffrement par bloc. Il peut supporter n'importe quelle combinaison de données (128 bits) et une taille de clé de 128, 192 et 256 bits. L'algorithme est appelé en fonction de la longueur de la clé AES-128, AES-192 ou AES-256. Pendant le processus de chiffrement-déchiffrement, l'algorithme AES passe par 10 tours pour une longueur de clé de 128-bits, 12 tours pour une longueur de clé de 192-bits et 14 tours pour une longueur de clé de 256 bits afin de fournir le texte chiffré ou pour récupérer le texte original (Guillot, 2013; Lafourcade, 2006; Thuillet, 2012).

AES divise les données de taille 128 bits en quatre blocs opérationnels de base. Ces blocs sont traités comme tableau d'octets et organisé sous forme d'une matrice de l'ordre 4 × 4 nommé l'état « states ».

Pour le processus de chiffrement et déchiffrement, le chiffrement commence par un stade d'AddRoundKey. Toutefois, avant d'atteindre le tour final, cette sortie passe par neuf tours principaux, au cours de chacun quelques quatre transformations sont accomplies (Guillot, 2013; Lafourcade, 2006; Thuillet, 2012):

1. La substitution des octets (Bute Substitution, BS): AES travaille sur des blocs de 128 bits ce qui signifie que chacun de ces blocs comporte 16 octets. Cette étape consiste à transformer chaque octet du bloc en un autre octet du bloc par l'utilisation d'une fonction non linéaire de substitution.
2. La rotation des lignes (Shift Row, SR): opère une transposition d'octet simple sur les lignes de l'état AES. La première ligne reste inchangée, mais les trois dernières sont décalées cycliquement en fonction de l'emplacement de la ligne. Pour la 2e ligne, un décalage circulaire à gauche de 1 octet est exécuté. Pour la 3 e et 4 e ligne, un décalage circulaire à gauche de 2 et 3 octets est effectué respectivement.
3. Le mélange des colonnes (Mix Column,MC): cette transformation correspond à une multiplication matricielle de chaque colonne des états. Chaque vecteur de colonne est multiplié avec une matrice de constantes. Dans cette opération, les octets sont pris comme des polynômes plutôt que des nombres.
4. L'addition des sous-clés (Add Sybkey): cette étape consiste à faire un « XOR » bit à bit entre l'état actuel et la clé du tour.

Dans le tour final (10), il n'y a pas de transformation du mélange des colonnes. Le schéma suivant illustre le fonctionnement de l'algorithme AES.

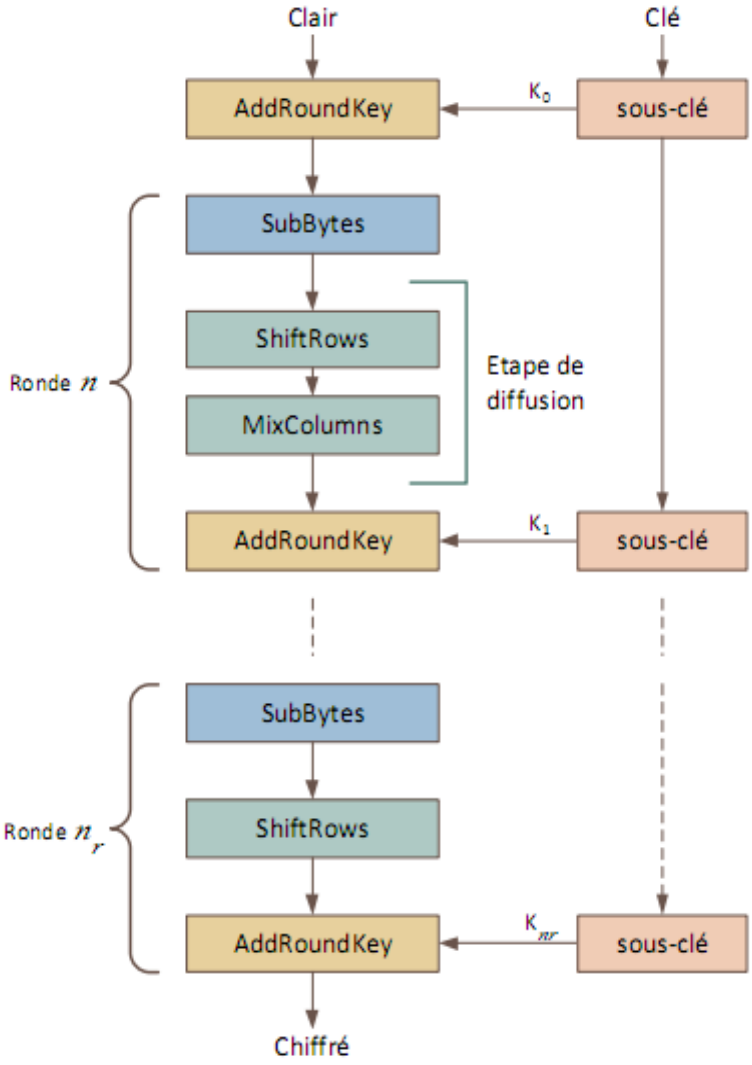

Figure 1.7 Chiffrement AES (Thuillet, 2012)

1.3.2. Chiffrement asymétrique

Contrairement au chiffrement symétrique qui utilise la même clé pour le chiffrement et le déchiffrement, le chiffrement asymétrique utilise deux clés différentes mais liées (clé publique et clé privée) ; l'une des clés est utilisée pour chiffrer et l'autre pour déchiffrer le message. Si la clé A est utilisée pour chiffrer un message, seule la clé B peut le déchiffrer, et si la clé B est utilisée pour chiffrer un message seule la clé A peut le déchiffrer. Le but principal de chiffrement asymétrique est de fournir des

solutions élégantes aux problèmes de partage de la clé secrète et la vérification d'identité. Le schéma de chiffrement asymétrique est illustré dans la Figure 1.8.

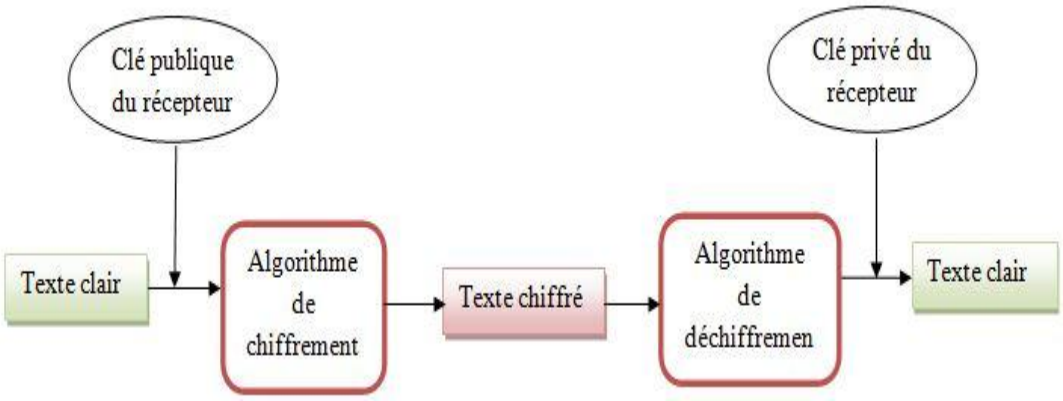

Figure 1.8 Chiffrement asymétrique

Nous détaillons ci-dessous les différentes méthodes de chiffrement asymétrique le plus utilisé ces dernières années.

1.3.2.1. Diffie-Hellman

Diffie-Hellman est le premier algorithme publié en 1976 par Whitfield Diffie et Martin E. Hellman de l'université de Stanford. Le but de cet algorithme est de résoudre le problème d'échange de la clé secrète entre deux communicants qui ne sont jamais rencontrés. Cet algorithme est basé sur deux problèmes, le premier est le DLP (Discrete Logarithm Problem) et le deuxième, le DHP (Diffie-Hellman Problem) (Guillot, 2013; Mansour, 2013).

- Le problème du logarithme discret (DLP): est une application réciproque de l'exponentiation, C'est l'analogue du logarithme qui est la réciproque de l'exponentielle. Nous présentons ci-dessous le DLP.

Instance: Soient G un groupe fini d'ordre n et g ϵ G. Etant donné un élément $\beta \epsilon$ G, Question: trouver le nombre entier a où $0 \leq a \leq n-1$, de telle sorte que $g^a = \beta$?

- Le DHP (Diffie-Hellman Problem): La motivation de proposer ce problème mathématique par Diffie et Hellman est parce qu'il existe de nombreux systèmes

de sécurité utilisant des opérations mathématiques qui sont rapides à calculer, mais difficiles à inverser. Le problème Diffie-Hellman est représenté de façon informelle comme suit. Il est très difficile de calculer g^{ab} par la connaissance de g, g^a et g^b (Guillot, 2013; Mansour, 2013).

> Étant donné un élément g et les valeurs de g^a et g^b, qu'elle est la valeur de g^{ab}? a et b sont des nombres entiers aléatoire et g est un générateur d'un groupe fini G.

Le protocole d'échange de clé Diffie-Hellman basé sur DLP et DHL sert à partager une clé secrète entre deux communicantes via un canal non sûr. Cette clé secrète peut être utilisée dans un procédé de chiffrement symétrique. La description de protocole Diffie-Hellman à clé publique est illustrée dans la figure suivante.

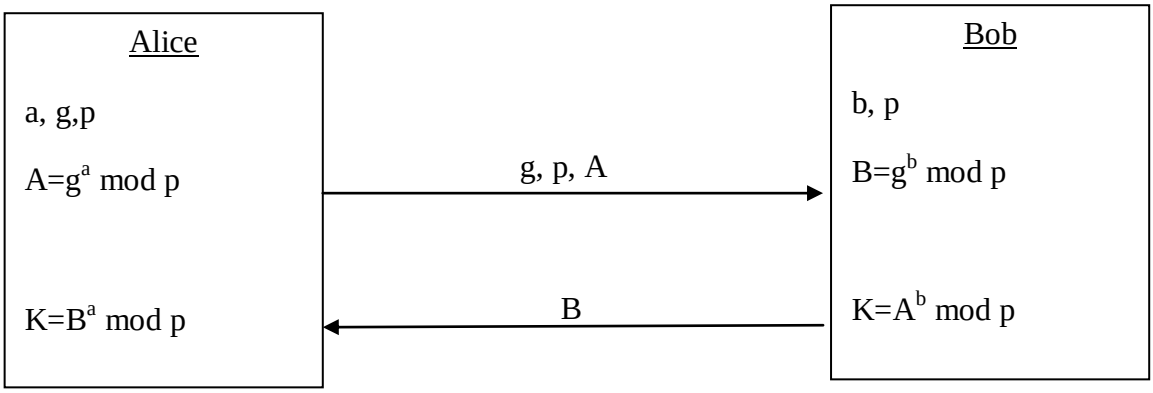

Figure 1.9 Protocole Diffie-Hellman

Les deux interlocuteurs (Alice et bob) partagent deux paramètres non secrets : un groupe fini cyclique G d'ordre p et un générateur g de G. Ensuite, chacun d'eux choisit un nombre secret de façon aléatoire (a et b respectivement), puis Alice et Bob calculent et échangent entre eux les valeurs publiques A et B tels que: $A= g^a \bmod p$ et $B= g^b \bmod p$. Alice et Bob, maintenant, peuvent calculer la clé secrète K telle que $K= g^{ab} \bmod p$.

1.3.2.2. Chiffrement RSA

RSA a été inventé en 1978 par Rivest, Shamir et Adelman, il se repose sur la difficulté de factoriser des produits de grands nombres premiers. La procédure de chiffrement est comme suit (Guillot, 2013; Mansour, 2013):

1. la généralisation de la clé publique et la clé privée: le récepteur effectue les opérations suivantes :
   - Choisit de façon aléatoire deux nombre premiers p et q;
   - Calcule n= p.q et ϕ (n) = (p-1) (q-1);
   - Choisit e tel que e est premier avec ϕ (n);
   - Choisit d tel que : e*d= 1 mod (ϕ (n));
   - La clé publique du récepteur est le couple (e, n) et sa clé privée est le couple (d, n).
2. Quand l'initiateur « Alice » veut envoyer un message « M » au répondeur « Bob », Alice doit chiffrer M par la clé publique de Bob tel que $c= m^e \bmod n$.
3. A la réception de c, Bob le déchiffre avec sa clé privée tel que $m= c^d \bmod n$.

1.3.2.3. Chiffrement El-Gamal

Le Crypto-système d'El-Gamal, ou chiffrement El-Gamal est un algorithme de cryptographie asymétrique (ou à clé publique) inventé par T. El-Gamal en 1985 : il est fondé sur le problème du logarithme discret et l'échange de clés de Diffie-Hellman. Le Crypto-système d'El-Gamal est utilisé par le logiciel libre GNU Privacy Guard, de récentes versions de PGP, et d'autres systèmes de chiffrement. Quand Alice veut envoyer un message chiffré à Bob par la méthode de chiffrement El Gamal, il suit le mode de fonctionnement suivant (El-Gamal, 1985; Guillot, 2013 Mansour, 2013):

- Alice et Bob partagent deux paramètres non secrets : un groupe fini cyclique G et un générateur g de G.
- Bob choisit un nombre premier p de telle sorte que le logarithme discret est incalculable.
- Bob choisit un nombre aléatoire k_b tel que $k_b < p$ et il calcule sa clé publique $B=g^{k_b} \bmod p$.
- Alice choisit un nombre aléatoire k et calcule $A=g^k$ et il chiffre le message m par $m' = m. B^k$, puis il transmet le couple (A, m') à Bob
- Pour déchiffrer le message chiffré d'Alice, Bob effectue l'opération suivante :

$$m = \frac{m'}{A^{k_b}} = \frac{m.B^K}{A^{k_b}} = \frac{m.g^{k_b k}}{g^{k k_b}}$$

1.3.2.4. Signature numérique

La signature numérique est une méthode qui sert à prouver l'identité de l'expéditeur d'un message, garantir l'intégrité des données et assurer la non-répudiation afin d'empêcher l'expéditeur de prétendre qu'il n'a pas envoyé les informations. Le schéma de signature numérique est basé sur la clé privée et la fonction de hachage.

Les fonctions de hachage ou les fonctions à sens unique, sont la quatrième primitive cryptographique. Une fonction de hachage est un algorithme qui prend en entrée un message de taille quelconque et applique un ensemble de transformations pour qu'il revoie en sortie un texte de taille fixe qui varie selon l'algorithme appliqué (128 bits pour MD5 et 160 bits pour SHA-1). Cette sortie est appelée le condensé, valeur de hachage ou résumé de message. Une fonction de hachage devrait satisfaire les quatre propriétés suivantes (Guillot, 2013; Mansour, 2013; Peyrin, 2008; Thuillet, 2012):

- facilité de calculer l'empreinte du message;
- impossible de reconstruire un message d'un condensé donné;
- impossible de trouver deux messages différents avec le même condensé;
- La taille du condensé (par une fonction de hachage donnée) est toujours la même (indépendamment de la longueur du message initial).

Pour signer un document, on devrait suivre les deux étapes suivantes. La première consiste à calculer le condensé d'un message afin d'assurer que le message ne sera pas altéré. La deuxième étape consiste à générer la signature numérique, le signataire prend le résultat de la première étape qui est le condensé et le chiffre avec sa clé privée. Ensuite, le signataire attache le résultat de chiffrement avec le texte clair et l'envoie au destinataire. La Figure 1.10 illustre le processus de génération de signature (Guillot, 2013; Mansour, 2013; Peyrin, 2008; Thuillet, 2012).

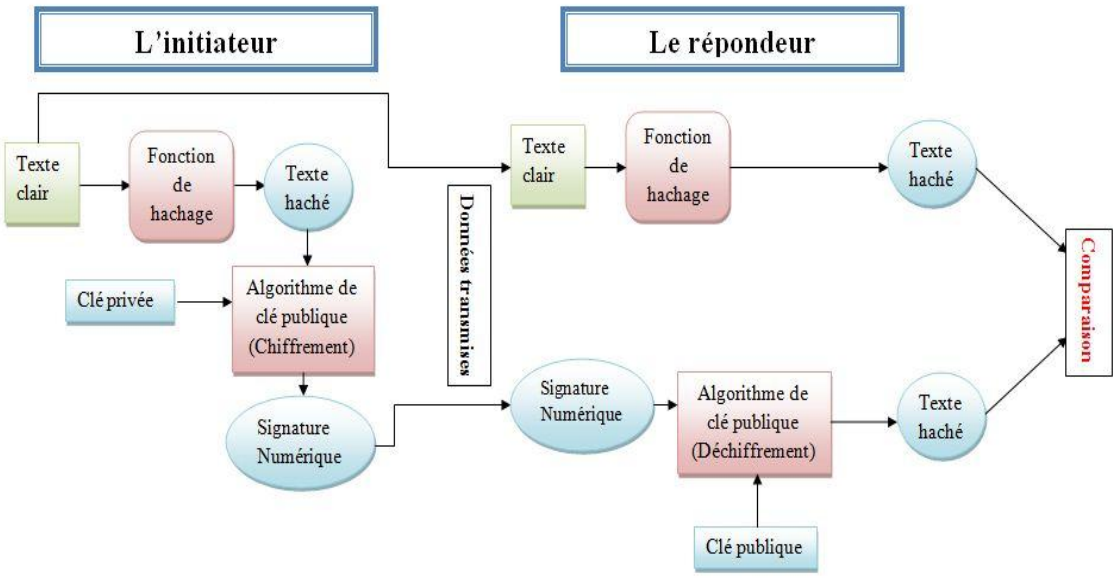

Figure 1.10 Signature numérique

1.3.2.4.1. Signature RSA

La signature RSA consiste à appliquer la méthode de RSA décrite dans la sous-section 1.3.2.2. Elle utilise la clé privée pour créer la signature et la clé publique pour la vérification de signature (Guillot, 2013; Mansour, 2013).

- **L'algorithme de signature RSA:**
  1. Calcule h = H (m) où H est une fonction de hachage;
  2. Calcule $S = h^d \bmod n$;
  3. Retour (s).
- **La vérification de signature de RSA:**
  1. Calcule h = H (m);
  2. Calcule $h' = s^e \bmod n$;
  3. Accepte la signature si et seulement si h = h'.

1.3.2.4.2. Signature El -Gamal

Supposons qu'Alice veut envoyer un message signé m à Bob par utilisation de l'algorithme El-Gamal. Tout d'abord, Alice et Bob se sont mis d'accord sur un groupe cyclique G, un générateur g de G, un grand nombre premier p et une fonction de hachage h (Guillot, 2013; Mansour, 2013).

- **Algorithme de signature:** Alice effectue les opérations suivantes:
  1. Choisit un nombre aléatoire k < p-1 tel que gcd (k, p-1) = 1;

2. Calcule : $r=g^k \bmod p$, $s= (h(m)- k_A\ r)k^{-1} \bmod p-1$ tel que k_A la clé privée de Alice;
3. Envoie (r, s) à Bob.

❖ **Algorithme de vérification :** à la réception du message d'Alice, Bob effectue la vérification suivante: $0 < r < p$, $0 < s < p$ et $g^{h(m)}= P_A\ r^s \bmod p$ tel que P_A la clé publique de Alice.

1.3.2.4.3. Digital Signature Algorithm (DSA)

L'algorithme de signature numérique DSA est un algorithme de signature numérique normalisé par NIST américain en août 1991. Le DSA peut être considéré comme une variante du schéma de signature El-Gamal, sa sécurité est basée sur le problème du logarithme (Guillot, 2013; Mansour, 2013).

❖ **Les paramètres publics de cet algorithme:** un grand nombre premier p, un diviseur q de p-1 de 160 chiffres binaires et un élément g de l'ensemble des entiers modulos p, différent de 1 et tel que $g^q = 1$.

❖ **Génération de clé:** on suppose qu'Alice veut envoyer un message signé à Bob. Alice effectue les opérations suivantes :

1. Choisit un nombre aléatoire ou pseudo-aléatoire X tel que $1 \leq x \leq q-1$;
2. Calcule $y = g^x \bmod p$;
3. La clé publique de Alice est y et sa clé privée est x.

❖ **Génération de la signature DSA:** pour signer un message m, Alice effectue les opérations suivantes :

1. Sélectionne un entier aléatoire ou pseudo-aléatoire k, $1 \leq k \leq q-1$;
2. Calcule $X = g^k \bmod p$ et $r = X \bmod q$. Si r = 0 alors on passe à l'étape 1;
3. Calcule $k^{-1} \bmod q$;
4. Calcule e = H(m);
5. Calcule $s = k^{-1}\{e+xr\} \bmod q$. Si s = 0 alors on passe à l'étape 1;
6. La signature de message est constituée de (r, s).

❖ **La vérification de la signature DSA:** pour vérifier la signature envoyée par Alice du message m, Bob effectue les opérations suivantes:

1. Vérifie que r et s sont des nombres entiers dans l'intervalle [1, q-1];
2. Calcule e = H(m);
3. Calcule $w = s^{-1} \bmod q$;

4. Calcule u_1 = e.w mod q et u2 = rw mod q;
5. Calcule X = $g^{u1}.y^{u2}$ mod p and v = X mod q;
6. Accepte la signature si et seulement si v = r.

1.3.2.5. Les courbes elliptiques

Les systèmes cryptographiques à courbes elliptiques (ECC) ont été inventés indépendamment par Miller et Koblitz en 1985. ECC fournit un niveau de sécurité équivalent à celui des autres protocoles asymétriques tels que : RSA, algorithme de signature numérique (DSA) ou Diffie-Hellman avec moins de tailles de clés. Grâce à l'utilisation des clés de petite taille, les systèmes ECC possèdent plusieurs avantages tels que la rapidité de calcul, une consommation d'énergie faible ainsi que l'économisassions de bande passante, ce qui rend les systèmes ECC utiles pour les réseaux où les ressources (puissance, mémoire, fréquence, bande passante, etc.) sont limitées (Blake, 2005; Guillot, 2013; Nitaj, 2011; Peyrin, 2008).

La base de tout système cryptographique à clé publique est un problème mathématique qui est mathématiquement impossible à résoudre. Par exemple RSA et Diffie-Hellman comptent sur la dureté de la factorisation des entiers et le problème du logarithme discret respectivement. Contrairement à la cryptographie à courbe elliptique, qui, elle repose sur des problèmes connexes dans le groupe des points d'une courbe elliptique sur un corps fini et le problème de logarithme discret.

Une courbe elliptique E (F_q) sur le champ F_q est un ensemble de points P = (P_x, P_y) qui satisfont un (Weierstrass) équation générale de la forme:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1.1)$$

Où a_1 a_2 a_3 a_4 a_6 ϵ F_q. Pour les applications cryptographiques, les courbes elliptiques sont définies sur un corps fini F_q où q est une puissance de nombre premier. a_1, a_2 et a_3 doivent avoir une valeur nulle et a_4 est devenu a et a_6 est devenu b, avec ces changements admissibles de variables dans l'équation précédente, on obtient une représentation plus familière:

$$y^2 = x^3 + ax + b,\ 4a^3 + 27\,b^2 \neq 0 \qquad (1.2)$$

Le noyau de fonctionnement de la courbe elliptique est une opération appelé par la multiplication scalaire, qui calcule Q = k.P (un point P multiplié k fois résultant dans un autre point Q sur la courbe). La multiplication scalaire est effectuée grâce à une combinaison d'ajouts de points. Pour plus de détail, voir les références Blake, 2005; Guillot, 2013; Nitaj, 2011; Peyrin, 2008). Nous présentons ci-dessus l'échange de clés (protocole Diffie et Hellman), l'algorithme de chiffrement et la signature numérique par courbes elliptiques.

1.3.2.5.1. Echange de clé de Diffie et Hellman

Le protocole d'échange de clé ECDH est similaire au protocole Diffie et Hellman en se basant sur les courbes elliptiques. Pour mieux expliquer le protocole ECDH on suppose qu'Alice et Bob veulent avoir une clé commune pour échanger des informations. Les étapes ci-dessous décrivent en détail les échanges effectuées entre Alice et Bob.

1. Alice et Bob s'accordent pour choisir une courbe elliptique E définie sur F_q sur laquelle le problème du logarithme discret est difficile à résoudre et un point $P \epsilon E(F_{q)}$ ayant un grand ordre;
2. Alice choisit un entier secret k_A et calcule le point $Q_A = k_A.P$ puis elle envoie Q_A à Bob;
3. Bob choisit un entier secret k_B et calcule le point $Q_B = k_B.P$ et il envoie Q_B à Alice;
4. Alice calcule: $k_A.Q_B = k_A.k_B.P$;
5. Bob calcule: $k_B.Q_A = k_B.k_A.P$;
6. La clé commune entre Alice et Bob est $k_A.Q_B = k_B.Q_A = k_A.k_B.P = k_B.k_A.P$.

1.3.2.5.2. Schéma de chiffrement avec les courbes elliptiques

Nous présentons dans cette sous-section les procédures de cryptage et de décryptage pour algorithme d'El-Gamal avec les courbes elliptiques. Supposons qu'Alice veut envoyer un message chiffré à Bob. Tout d'abord, Alice et Bob s'accordent pour choisir une courbe elliptique E définie sur F_q sur laquelle le problème du logarithme discret est difficile à résoudre et un point $P \epsilon E(F_q)$ ayant un grand ordre. Puis Alice et Bob accomplissent les opérations suivantes (Mansour, 2013):

1. Alice choisit un entier aléatoire $k_A \in E(F_q)$ qui représente sa clé privée et calcule $Q_A = k_A.P$ qui représente sa clé publique;
2. Bob choisit un entier aléatoire $k_B \in E(F_q)$ qui représente sa clé privée et calcule $Q_B = k_B.P$ qui représente sa clé publique;
3. Alice effectue les opérations suivantes:
   - Représente le message m comme point M dans $E(F_q)$;
   - Sélectionne $k \in R [1, n-1]$;
   - Calcule $C1 = M + k_A.Q_B$;
   - Envoie $(Q_A, C1)$ à Bob.
4. A la réception de message d'Alice, Bob effectue les opérations suivantes: Calcule : $C1 - k_B.Q_A = M + k_A.Q_B - k_B.k_A.P = M + k_A.k_B.P = M$.

1.3.2.5.3. Schéma de la signature avec les courbes elliptique

❖ **ECDSA (Elliptic Curve Digital Signature Algorithm)**

ECDSA (Elliptic Curve Digital Signature Algorithm) est une variante de l'algorithme de signature électronique qui fonctionne sur les courbes elliptiques. ECDSA a été proposé par Scott Vanstone en 1992, en réponse à un appel d'offres pour les signatures numériques du NIST (National Institute of Standards and Technology). Dans ce qui suite, nous présentons le déroulement de l'algorithme ECDSA qui se compose de deux processus: la génération de signature et la vérification de signature.

Soient une courbe elliptique E définie sur un corps fini F_q, un point sur la courbe d'ordre $G \in E(Fq)$ et une fonction de hachage H. On suppose que l'expéditeur Alice et le récepteur Bob sont deux parties de communicantes, et Alice veut envoyer un message signé à Bob (Mansour, 2013).

- **La génération de la paire de clés:** Alice effectue les opérations suivantes:
  1. Sélectionne un nombre aléatoire ou pseudo-aléatoire dans l'intervalle [1, n-1];
  2. Calcule Q = d.G;
  3. La clé publique d'Alice est Q et sa clé privée est d.
- **Algorithme de la génération de la signature:** pour générer la signature du message, Alice effectue les opérations suivantes:

1. Sélectionne un nombre aléatoire ou pseudo-aléatoire entier k, $1 \leq k \leq n-1$;
2. Calcule k.G = (x1, y1);
3. Calcule r = x1 mod n. Si r = 0 alors Alice retourne à l'étape 1;
4. Calcule k^{-1} mod n;
5. Calcule e=H(m);
6. Calcule $s = k^{-1}(e + dr) \bmod n$. Si s = 0 alors Alice retourne à l'étape 1;
7. La signature de message m est égale à (r, s).

- **<u>Vérification de la signature:</u>** pour vérifier la signature envoyée par Alice, Bob effectue les opérations suivantes:
  1. Vérifie que r et s sont des nombres entiers dans l'intervalle [1, n-1];
  2. Calcule e=H (m);
  3. Calcule $w = s^{-1} \bmod n$;
  4. Calcule u1 = e.w mod n et u2 = r.w mod n;
  5. Calcule X = u1.G + u2.Q;
  6. Si X = O, Bob rejette la signature. Sinon, Bob convertit le x1 coordonné de X en x1 entier, et calcule v = x1 mod n;
  7. Accepte la signature si et seulement si v = r.

❖ **Signature d'El-Gamal**

Alice veut envoyer un message m signé par la signature d'El-Gamal à Bob. Pour cela Alice choisit une courbe elliptique E définie sur un corps fini F_q, une fonction de hachage H et une fonction f : E(Fq) → Z. Elle choisit un grand nombre premier A ϵ $E(F_q)$ et un nombre secret a, puis calcule B = a.A (Mansour, 2013).

- **<u>Génération de la signature:</u>** pour signer le message, Alice effectue les opérations suivantes :
  1. Choisit un nombre entier k avec PGDC (k,n) = 1;
  2. Calcule R = k.A;
  3. Calcule $s \equiv k^{-1} (H(m) - a.f(R)) \pmod n$;
  4. Le message signé est : (m,s,R).
- **<u>Vérification de la signature :</u>** pour vérifier la signature d'Alice, Bob effectue les opérations suivantes:
  1. Télécharge l'information publique d'Alice;
  2. Calcule V1=f(R).B+s.R et V2= H(m).A;

3. Si V1=V2 alors la signature est valide.

1.3.3. Etude comparative

Dans cette section nous présentons une comparaison entre les algorithmes de chiffrement symétrique/asymétrique qui sont complémentaire dans les protocoles de sécurité (voir tableau 1.1). Une autre comparaison entre un système asymétrique basé sur RSA et un système asymétrique basé sur les courbes elliptique a été réalisée par NIST. Le tableau 1.2 ci-dessous est publié par le NIST, il définit les forces de sécurité équivalentes entre le RSA, ECC, et des algorithmes de clés symétriques.

Les systèmes de chiffrement	**Symétrique**	**Asymétrique**
Les avantages	- La rapidité de calcule - La taille de clé utilisé pour le chiffrement et déchiffrement est courte - Facile à implémenter au niveau matérielle parce qu'il se base sur des opérations simple.	- Aucun canal secret n'est nécessaire pour l'échange des clés publiques. - Nécessite que 2×n clé pour n entité de communication. - Assure la signature de message
Les inconvénients	- Nécessite un canal secret pour envoyer la clé secrète - Il ne garantit pas la propriété de non-répudiation. - Le changement de clé à chaque communication. - Le nombre de clé à gérer égale à n×(n-1) ⁄ 2, tel que n est le nombre des entités communicante .	- La taille de clé utilisé pour le chiffrement et le déchiffrement est très grande. - Très lent par apport au chiffrement symétrique - Nécessite des machines puissantes.

Tableau 1.1 Comparaison entre les systèmes de chiffrement symétrique et asymétrique

Symmetric Key Size (m)	Symmetric Algorithm	Hash Algorithm	Elliptic curve modulus	RSA Modulus	Expected Lifetime
56 Bits	DES	-	-	-	Expired
60 Bits	-	MD5	111 Bits	512 Bits	Expired
80 Bits	3DES (2 Key)	SHA-1	160 Bits	1024 Bits	2010
112 Bits	3DES (3 key)	SHA-224	224 Bits	2048 Bits	2030
128 Bits	AES-128	SHA-256	256 Bits	3072 Bits	2031+
192 Bits	AES-192	SHA-384	384 Bits	7680 Bits	2031+
256 Bits	AES-256	SHA-512	512 Bits	15360 Bits	2031+

Tableau 1.2 NIST Crypto Modernization Guidelines (NIST, 2013)

Par exemple, l'algorithme AES, avec une taille de clé 128 bits, exige une taille de clé 3072 bits de l'algorithme RSA, mais ne nécessite qu'une taille de clé de 256 bits pour un algorithme basé sur ECC pour une même force de sécurité.

1.4. Conclusion

De nos jours, la sécurité des réseaux informatique est devenue de plus en plus importante. Dans le but de protéger les réseaux informatiques contre les différents types d'attaque, différents techniques et mécanismes de sécurité ont été inventés, parmi ces techniques on trouve la cryptographie qui représente la base de tout mécanisme de chiffrement. Dans ce chapitre, nous avons présenté les dimensions de sécurité et les différents types d'attaque. Ainsi que les méthodes de chiffrements symétrique et asymétrique. Dans le chapitre suivant, nous décrivons les protocoles de sécurité qui utilisent ces derniers.

Chapitre II : Les protocoles de sécurité: SSH, SSL/TLS, IPSec

Après avoir introduit les éléments fondamentaux et les principes de base de la cryptographie, nous abordons le sujet réel de notre thèse. Dans ce chapitre, nous présentons les différentes technologies et les protocoles qui peuvent être utilisés pour fournir des services de sécurité pour les réseaux IP où nous nous focalisons sur les protocoles qui nous permettent de sécuriser les échanges au niveau de la couche réseau.

2.1. Les protocoles de sécurité

Vu que le format des paquets Internet est publiquement défini et bien connu, un paquet qui traverse l'Internet peut être capturé par l'un des routeurs qui se trouvent sur son chemin (Frankel, 2001). Donc, il est relativement facile de falsifier les adresses des paquets IP, modifier le contenu des paquets IP, rejouer un vieux paquet, et inspecter le contenu des paquets IP en transits. Par conséquent, rien ne garantit que les datagrammes IP reçus: provient du vrai expéditeur (l'adresse source dans l'entête IP); contiennent les vraies données expédiées; n'ont pas été inspectées par un autre tiers. Plusieurs protocoles de sécurité permettent de pailler ces faiblesses. Parmi ces protocoles on trouve, le SSL/TLS qui opère au niveau transport, le protocole SSH qui opère au niveau applicatif et le protocole IPSec qui opère au niveau réseau de pile protocolaire (Doraswamy and Harkin, 2003).

2.1.1. Le protocole SSL/TLS

Le protocole SSL (Socket Secure Layer) fonctionne suivant un mode client/serveur. Il permet d'assurer les services de sécurité suivante aux clients TCP: l'authentification du serveur; les services de confidentialité; l'intégrité des services de connexion (Lamotte et al., 2005; Oppliger, 2009).

Le protocole SSL a été développé par Netscape Communication en 1994. Il a directement publié la version SSLv2 en 1995. Un certain nombre de lacunes ont été découverte dans cette version. Pour cette raison, la version 3 a été publiée en novembre 1996 et a permet de pallier ces lacunes. En mai 1996, l'Internet Engineering Task Force (IETF) a mis en place un groupe de travail appelé Transport Layer Security (TLS) afin de poursuivre les travaux de Netscape Communication. En

janvier 1999 l'IETF a publié TLS 1.0 (considéré comme étant SSL 3.1) (Lamotte et al., 2005; Oppliger, 2009).

Les concepteurs de SSL ont décidé de créer un protocole distinct seulement pour la sécurité. En effet, ils ont ajouté une couche à l'architecture de protocole de l'Internet (IP). La figure 2.1 (Stephen, 2000) montre les principaux protocoles pour les communications Web. En bas on trouve le protocole Internet (IP). Ce protocole est responsable de l'acheminement des messages à travers des réseaux à partir de leur source jusqu'à leur destination. Au milieu on trouve le protocole TCP (Transmission Control Protocol) qui permet d'assurer la fiabilité de la communication. Au-dessus il opère l'HyperText Transfer Protocol (http) qui comprend les détails de l'interaction entre les navigateurs Web et les serveurs Web. Le SSL assure la sécurité en agissant comme un protocole de sécurité séparé, le SSL est introduit entre l'application HTTP et TCP comme le montre dans la figure 2.2 (Stephen, 2000).

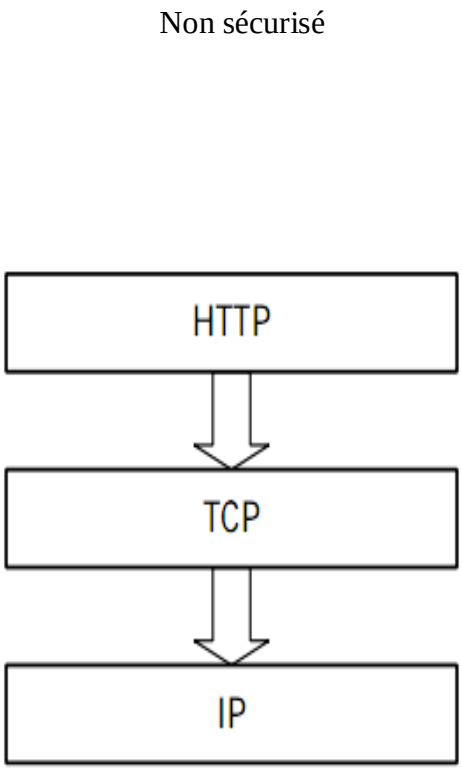

Figure 2.1 Les protocoles de communication

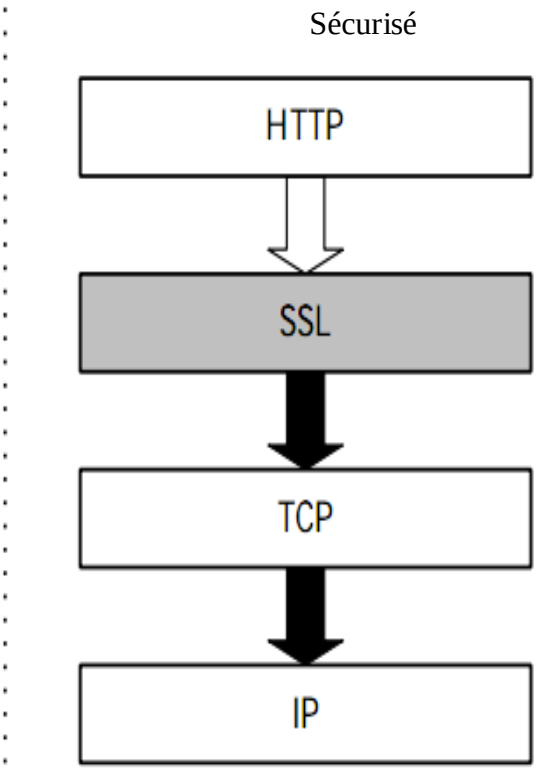

Figure 2.2 SSL est une couche de protocole particulier assurant la sécurité (Stephen, 2000)

2.1.1.1. Architecture du protocole SSL

Le protocole SSL se compose de plusieurs protocoles comme l'illustre la figure 2.3: le protocole d'enregistrement (Record protocol), le protocole de poignée de main (Handshake Protocol), le protocole de changement des spécifications de chiffrement (Change Cipher Spec Protocol), le protocole d'alerte (Alert Protocol) et le protocole de données d'application (Stephen, 2000).

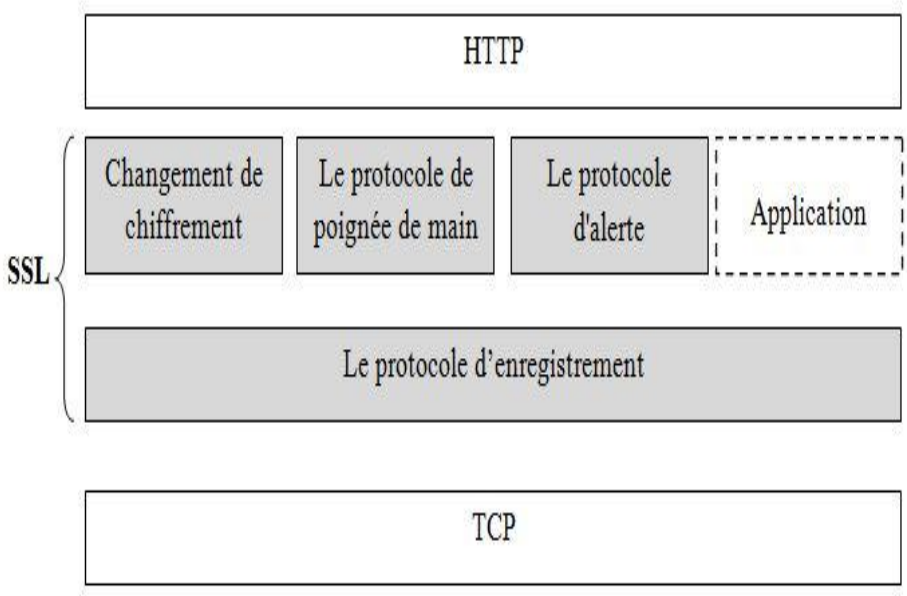

Figure 2.3 Les composants du protocole SSL (Stephen, 2000)

2.1.1.1.1. Le protocole d'enregistrement (Record protocol)

Le protocole d'enregistrement est utilisé pour l'encapsulation des données du protocole de la couche supérieure. Le traitement de protocole d'enregistrement est illustré à la figure 2.4. Il commence par la fragmentation des données de la couche supérieure en blocs de 214 octets ou moins (appelés des fragments). Ensuite, il compresse chaque fragment selon la méthode de compression spécifié dans l'état de la session SSL. Puis il calcule un condensât (MAC) et il protège le fragment compressé et le condensât selon l'algorithme de chiffrement spécifié dans l'état de la session SSL. Enfin, le protocole d'enregistrement SSL ajoute un en-tête d'enregistrement SSL au fragment crypté pour obtenir l'enregistrement SSL. L'entête de l'enregistrement SSL comprend les trois champs suivants (figure 2.5) (Oppliger, 2009):

1. Le type du contenu qui occupe les premier 8 bits : Il fait référence au protocole SSL de couche supérieure. Il y a quatre valeurs prédéfinies (Oppliger, 2009):
   - ❖ 20 : fait référence au protocole de changement des spécifications de chiffrement (Change Cipher Spec Protocol);
   - ❖ 21 : fait référence au protocole d'alerte ;
   - ❖ 22 : fait référence au protocole de poignée de main (Handshake Protocol);
   - ❖ 23 : se réfère au protocole des données d'application.
2. La version de protocole : se réfère à la version du protocole SSL utilisé. Il occupe les deux octets qui se composent de valeur major et mineure de numéro de version de SSL/TLS: 2.0 pour SSL, v2, 3.0 pour SSLv3 et 3.1 pour TLS.
3. Un champ d'une longueur de 16 bits qui se réfère à la longueur du fragment, chaque fragment est de taille 214 octet ou inferieure après la décompression.

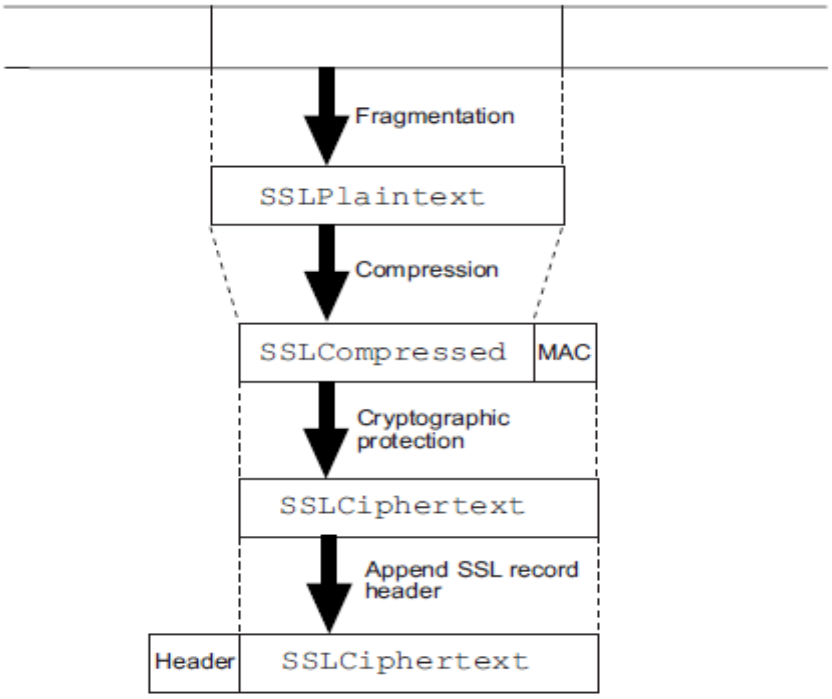

Figure 2.4 Le processus du protocole d'enregistrement SSL (Oppliger, 2009)

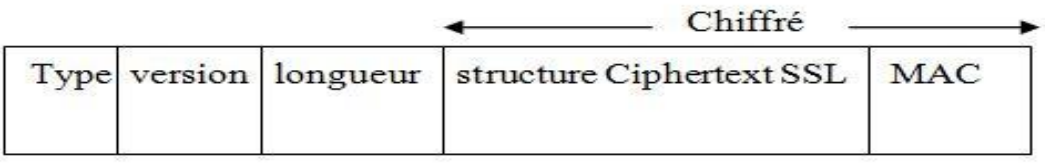

Figure 2.5 L'entête d'un enregistrement SSL (Oppliger, 2009)

2.1.1.1.2. Le protocole de poignée de main (Handshake Protocol)

Le protocole de négociation SSL permet à un client et un serveur de s'authentifier mutuellement, de négocier les paramètres confidentiels comme les algorithmes de chiffrement et les méthodes de compression et d'établir les clés qui seront utilisées dans les algorithmes choisis. La figure 2.6 illustre le format des messages de négociation à travers des enregistrements et indique que plusieurs messages de prise de contact peuvent être (et ils sont fréquemment) combinés en un message de couche d'enregistrement unique. La partie fortement encadrée d'un message fait référence au message (s) de négociation SSL, alors que les 5 principaux octets se réfèrent à l'entête d'enregistrement SSL. Cet entête, à son tour, comporte toujours une valeur de type (22) de 1 octet (se référant au Protocole SSL Handshake), une valeur de la version 3,0 de 2 octets, et une valeur de longueur 2 octets se référant à la longueur en octets de la partie restante du disque SSL (Oppliger, 2009).

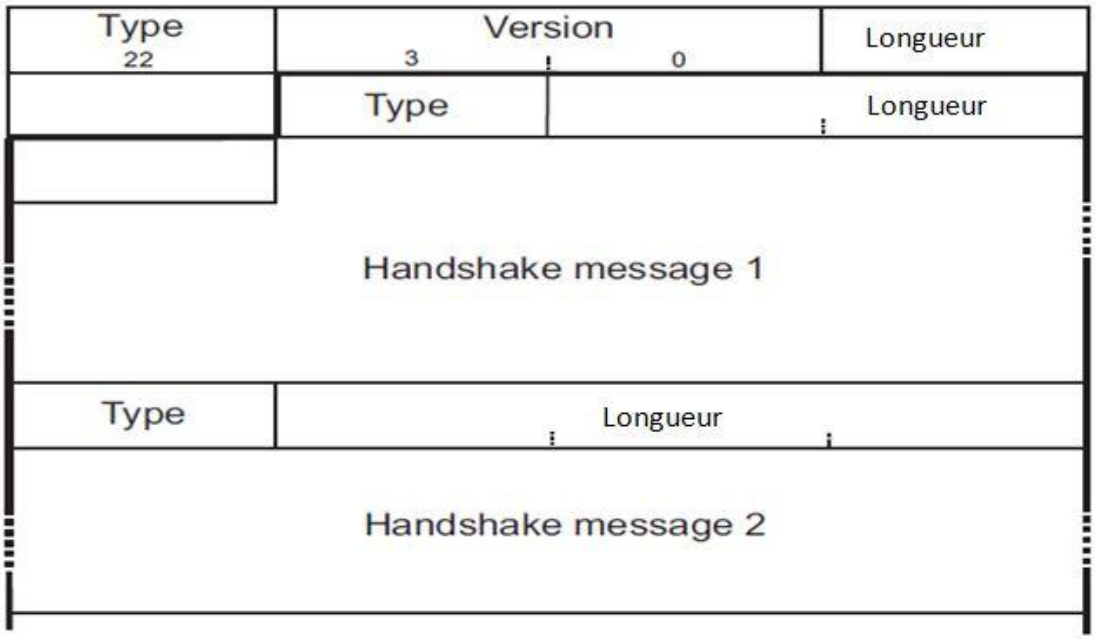

Figure 2.6 La structure d'un message de protocole de négociation SSL (Oppliger, 2009)

❖ **Les étapes de la négociation:**

Le protocole et les flux des messages sont illustrés à la figure 2.7. Les messages qui sont écrits entre crochets sont facultatifs ou dépendants de la situation, ce qui signifie

qu'ils ne sont pas toujours envoyés. Le protocole de négociation SSL comprend quatre ensembles de messages qui sont échangés entre le client et le serveur. Chaque ensemble est typiquement transmis dans un segment TCP séparé (Oppliger, 2009).

- Le premier ensemble de messages est envoyée par le client au serveur. Il comporte uniquement un message de « CLIENTHELLO » (type 1). Ceci indique que le client veut établir une connexion sécurisée (Oppliger, 2009).
- Le deuxième ensemble de messages comporte 2-5 messages qui sont envoyés par le serveur au client (Oppliger, 2009):
    1. Un message de « SERVERHELLO » (type 3) est envoyé comme réponse de message « CLIENTHELLO » pour lui indiquer qu'il a bien reçu son message.
    2. Si le serveur s'est authentifié (ce qui est généralement le cas), il peut envoyer un message de « CERTIFICATE » (type 11) au client.
    3. Dans certaines conditions le client demande de transmettre des clés sécurisées, le serveur peut envoyer un message de «SERVERKEYEXCHANGE » (type 12) au client.
    4. Si le serveur exige que le client s'authentifie avec un certificat, il peut envoyer un message de « CERTIFICATE REQUEST » (type 13) au client.
    5. Enfin, le serveur envoie un message de SERVERHELLODONE (type 14) au client.

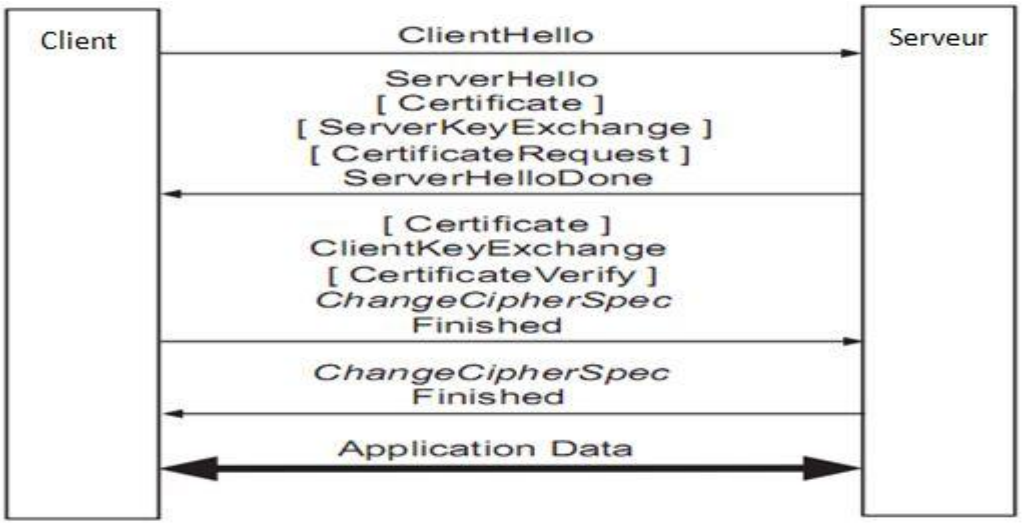

Figure 2.7 Le protocole de poignée de main (Handshake Protocol) (Oppliger, 2009)

A la fin des échanges des messages ClientHello et ServerHello, le client et le serveur achèveront la négociation d'une version de protocole et auront un identifiant de session (ID), l'algorithme de chiffrement, et une méthode de compression. En outre, deux valeurs aléatoires (ClientHello.random et ServerHello.random) sont générées et disponibles pour l'utilisation.

- Le troisième ensemble de messages comprend 3-5 messages qui sont de nouveaux envoyés à partir du client vers le serveur:
    1. Si le serveur envoie un message de « CERTIFICATE REQUEST », le client envoie un message de « CERTIFICATE » (type 11) au serveur.
    2. Dans l'étape principale du protocole, le client envoie un message de « CLIENTKEYEXCHANGE » (type 16) au serveur. Le contenu de ce message dépend de l'algorithme d'échange de clé en cours d'utilisation.
    3. Si le client a envoyé un certificat au serveur, il doit également envoyer un message « CERTIFICATEVERIFY » (type 15) au serveur. Ce message est signé numériquement avec la clé privée correspondant à la clé publique du certificat.
    4. Le client envoie un message « CHANGECIPHERSPEC » au serveur (en utilisant Le protocole de changement des spécifications de chiffrement).
    5. Le client envoie un message crypté « FINISHED » (type 20) au serveur.
- le quatrième ensemble de messages comprend deux messages qui sont envoyés par le serveur vers le client :
    1. Le serveur envoie un autre message « CHANGECIPHERSPEC » au client.
    2. le serveur envoie un message (crypté) « FINISHED» (type 20) au client.

A ce moment, la négociation SSL est terminée; le client et le serveur peuvent commencer à échanger les données de la couche application (en utilisant le protocole SSL Application Data) (Oppliger, 2009).

2.1.1.1.3. Le protocole d'alerte

Le protocole d'alerte est utilisé pour notifier des avertissements ou des erreurs qui auraient pu se produire, par exemple si un certificat ne pouvait pas être vérifié. Le protocole SSL permet d'alerter les pairs de communication pour échanger des

messages d'alerte. Chaque message d'alerte porte un niveau d'alerte et une description d'alerte (Oppliger, 2009):

- Le niveau d'alerte comprend 1 octet, où la valeur 1 signifie «avertissement» et la valeur 2 signifie « fatale ». Pour tous les messages d'erreurs pour lesquels le niveau d'alerte particulière n'est pas explicitement spécifié, l'expéditeur peut déterminer (d'après sa propre discrétion) si elle est fatale ou non. De même, si une alerte avec un niveau d'alerte d'avertissement est reçue, le récepteur peut décider à sa discrétion s'il doit le traiter comme une erreur fatale.
- La description de l'alerte comprend également 1 octet, où un code numérique se réfère à une situation spécifique (Oppliger, 2009).

2.1.1.1.4. Le protocole d'application des données

Le protocole d'application des données SSL permet aux pairs de communication d'échanger des données selon un protocole de couche d'application. Plus précisément, il prend les données d'application et il les transmet au protocole d'enregistrement SSL pour la fragmentation, la compression et la protection cryptographique (Oppliger, 2009).

2.1.2. Le protocole SSH (Secure Shell)

SSH (Secure Shell) est une approche populaire puissante basé sur un logiciel de sécurité du réseau. À chaque fois que les données sont envoyées par un ordinateur au réseau, SSH crypte automatiquement ces données. Lorsque les données atteignent leur destinataire, SSH décrypte ces données automatiquement. SSH utilise des algorithmes de cryptage modernes et il est suffisamment efficace pour être trouvé dans les applications à mission critique dans les grandes sociétés.

SSH-1 (Secure Shell) a été développé en 1995 par Tatu Ylönen, un chercheur dans laboratoire informatique à l'université d'Helsinki en Finlande. En Juillet 1995, la première version de SSH (SSH-1) a été publiée au public autant que un logiciel libre avec son code source, permettant aux gens de le copier et utiliser le programme sans frais dans le but de sécuriser les communications distantes. À cause de plusieurs faiblesses et limites découvertes dans la première version de SSH, l'IETF a formé un groupe de travail appelé SECSH (Secure Shell) pour normaliser le protocole et

guider son développement dans l'intérêt public. Le groupe de travail a présenté le premier SECSH Projet Internet pour le protocole SSH-2.0 en Février 1997 (Hajjeh, 2004; Stallings, 2014 ; Ylonen et al., 2006a).

2.1.2.1. L'architecture et le fonctionnement de base du protocole SSH

Le protocole SSH est un protocole applicatif (la couche 7 du modèle OSI) qui permet d'établir une connexion sécurisée entre un client SSH et un serveur SSH à distant. Il assure l'authentification, le chiffrement et l'intégrité des données transmises dans un réseau. Dans ce qui suit, nous présentons l'architecture du protocole SSH-2 (Figure 2.8). Ce protocole est subdivisé en trois protocoles : SSH Transport Layer Protocol (SSH-TRANS) (Ylonen et al., 2006b) ; SSH Authentication Protocol (SSH-AUTH) (Ylonen et al., 2006c); SSH Connection Protocol (SSH-CONN) (Ylonen et al., 2006d).

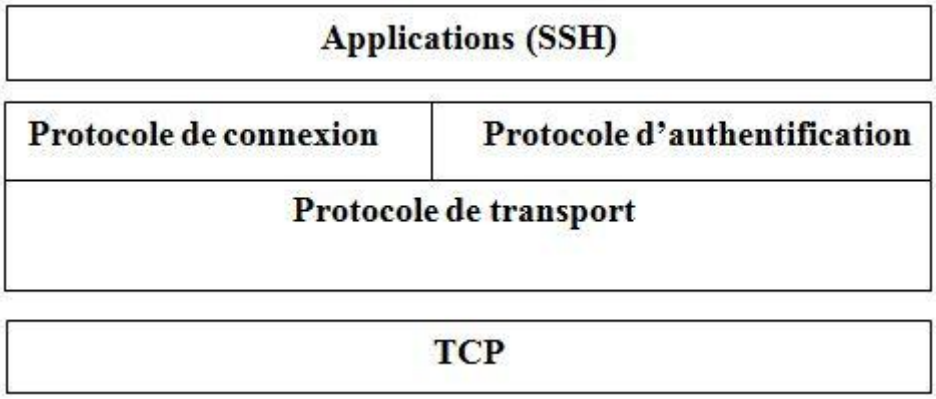

Figure 2.8 L'architecture du protocole SSH-2 (Hajjeh, 2004)

- SSH Transport Layer Protocol (SSH-TRANS): définit le protocole de la couche transport. Ce protocole fournit un canal confidentiel sur un réseau non sécurisé. Il effectue l'authentification du serveur, l'échange de clés, le chiffrement, la protection de l'intégrité et la compression. Il tire aussi un identifiant de session unique qui peut être utilisé par les protocoles du niveau supérieur. La couche de transport sera généralement exécutée sur une connexion TCP / IP, mais peut aussi être utilisée au-dessus de tout autre flux de données fiable (Ylonen et al., 2006d).
- SSH Authentication Protocol (SSH-AUTH): définit le protocole d'authentification. Ce protocole fournit un ensemble des mécanismes qui peuvent être utilisés pour authentifier le client pour le serveur. Les mécanismes

individuels précisés dans le protocole d'authentification utilisent l'identifiant de la session fournie par le protocole de transport (Ylonen et al., 2006d).

- SSH Connection Protocol (SSH-CONN) : définit le protocole de connexion, ce protocole permet de multiplexer de multiples canaux de communication logiques sur une seule connexion SSH sous-jacente (Ylonen et al., 2006d).

2.1.2.1.1. La phase d'initialisation du protocole

Pendant la phase d'initialisation du protocole SSH (figure 2.9), la procédure de négociation des informations entre la machine cliente et la machine serveur se déroule comme suit (Bouamama et al., 2008; ISS, 2008):

1. Le client et serveur se mettent d'accord sur la version du protocole SSH;
2. Le serveur envoie au client la liste des méthodes d'authentification proposées, la liste des algorithmes de chiffrements proposés, des indicateurs d'extensions du protocole (par exemple, la méthode de compression, etc.) et un cookie codé sur 64 bits dans le but de protéger le serveur contre l'attaque DoS;
3. Le client envoie au serveur une copie du cookie, la liste des algorithmes de chiffrement sélectionnés et la liste des méthodes d'authentification sélectionnées;
4. Le client et le serveur sélectionnent les meilleurs algorithmes parmi les algorithmes proposés;
5. Le client et le serveur échangent les valeurs de Diffie-Hellman; puis, ils calculent un identifiant de session à partir de ces valeurs;
6. Le serveur envoie au client sa clé publique et il signe les valeurs échangées précédemment avec sa clé privée;
7. Le client passe en mode crypté après la vérification de la signature du serveur;
8. Le serveur envoie au client un message de confirmation crypté;
9. Les deux entités passent en mode crypté ;
10. Le client envoie au serveur la demande d'un service;
11. Le serveur sélectionne les méthodes d'authentification;
12. Finalement, le client envoie au serveur la méthode d'authentification choisie qui peut être acceptée ou rejetée par le serveur (Bouamama et al., 2008; ISS, 2008).

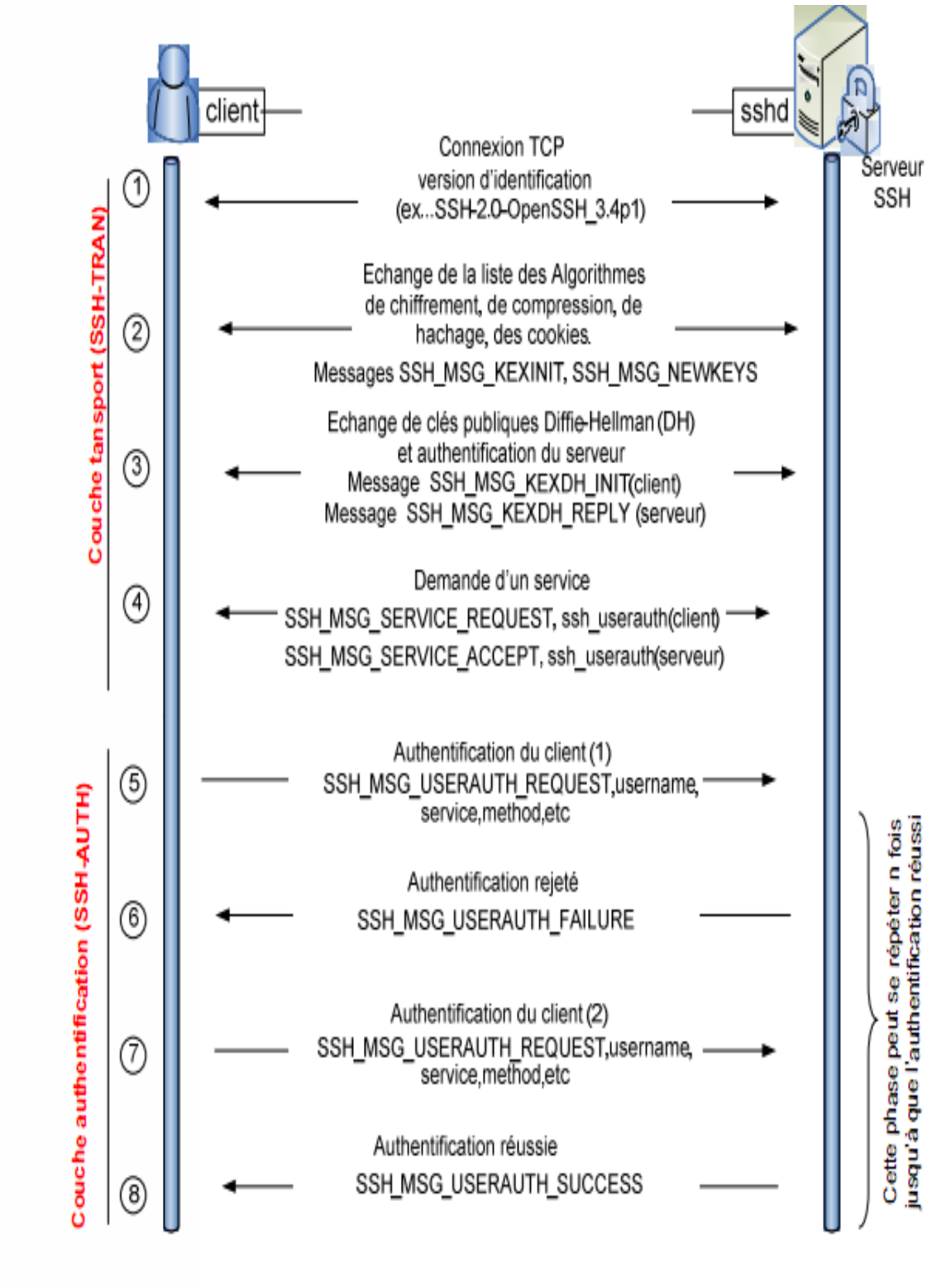

Figure 2.9 La phase d'initialisation du protocole SSHv2 (ISS, 2008)

2.1.2.1.2. Les méthodes d'authentification utilisée dans la version 2 normalisée par l'IETF

Le protocole SSH-2 supporte plusieurs méthodes d'authentification avec des propriétés de sécurité différentes (ISS, 2008).

- Password
  - Il s'agit de l'authentification classique, le client envoie d'une manière sécurisée au serveur son mot de passe;
  - Le serveur récupère le mot de passe et calcule son hash; puis, il compare le hash calculé avec l'empreinte du mot de passe du client stocké dans sa base de données;
  - Notons que:
    - Dans le système Unix, les mots de passe stockés dans la base de données sont chiffrés à l'aide de l'algorithme DES (via la fonction crypt()).
    - D'autres systèmes utilisent des fonctions de hachage telles que MD5 ou SHA-1 (ISS, 2008).
- Publickey
  - L'authentification à clé publique est basée sur la cryptographie asymétrique (RSA ou DSA) où aucun secret ne circule sur le réseau. En effet, la clé publique du client doit être stockée sur le serveur SSH et sa clé privée doit être stockée sur sa machine d'une manière sécurisée;
  - Publickey (RSA ou DSA) possède un niveau de sécurité plus élevé que le système par mot de passe. (ISS, 2008).
- Hostbased
  - Est une méthode d'authentification identique à celle utilisée par les r-commandes et les fichiers tels que /etc/rhosts et ~/.rhosts, qui « certifient » les sites clients ayant préalablement enregistré leur adresse dans le serveur;
  - Dans cette méthode d'authentification, lorsque le client envoie une demande de connexion à un serveur SSH, ce dernier opère une recherche dans les fichiers rhosts afin de trouver le nom de l'hôte qui correspond à l'adresse source de la connexion réseau du client (ISS, 2008).

2.1.3. Internet Protocol Security (IPSec)

Internet Protocol Security (IPSec) est un cadre de protocole open-source pour le développement de la sécurité au sein de la famille protocole TCP / IP (Figure 2.10). Il est utilisé pour sécuriser les communications sur les réseaux IP tels que LAN, WAN. Il a été définit par un groupe de travail de IETF (Internet Engineering Task Force) depuis 1992. En 1995, une première version a été développée sous forme de RFC (Request For Comment), sans la partie de gestion des clés. Une amélioration de protocole a été faite en 1998, elle permet d'ajouter un système dynamique pour la gestion des paramètres confidentielle de ce protocole (Labouret, 2000; Michael and Herbert, 2009).

Le protocole IPSec prend en charge l'intégrité des données, la confidentialité des données, l'authentification de l'origine des données, et la protection contre le rejeu au niveau du réseau informatique. Puisque IPSec est intégré à la couche Internet (couche 3), il assure la sécurité pour tous les protocoles de la suite TCP / IP, et vu que IPSec est appliquée de manière transparente aux applications, il n'est pas nécessaire de configurer la sécurité distincte pour chaque application qui utilise le protocole TCP / IP (Michael and Herbert, 2009).

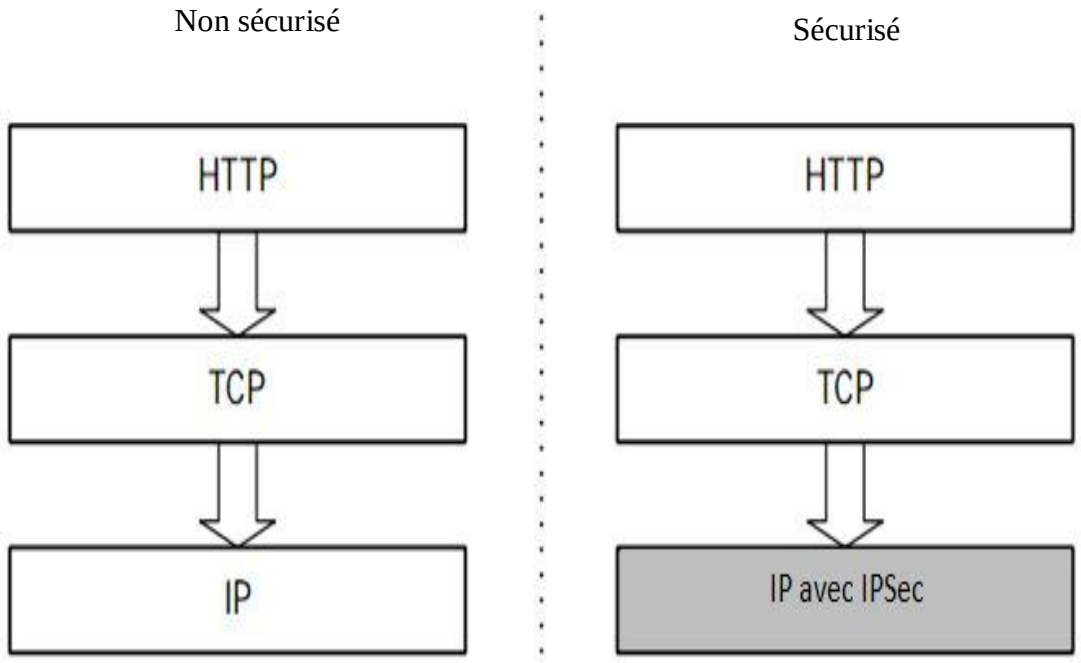

Figure 2.10 La sécurité du réseau avec IPSec (Stephen, 2000)

2.1.3.1. Les services offerts par IPSec

Les protocoles d'IPSec (AH et ESP) peuvent être utilisés pour protéger soit une charge utile IP-complète ou seule la charge utile IP des protocoles des couches supérieures. Cette distinction est traitée en considérant deux modes différents d'IPSec.

2.1.3.1.1. Mode transport

Le mode transport est le mode par défaut d'IPSec, il est utilisé pour les communications de bout en bout (par exemple, pour les communications entre un client et un serveur). En mode transport, un entête d'IPSec est inséré entre l'entête IP et l'entête de protocole de la couche supérieure, seules les données provenant de cette couche sont cryptées. La figure 2.11, illustre les paquets IP protégés par AH-IPSec et ESP-IPSec en mode transport (Doraswamy and Harkin, 2003 ; Kozierok, 2005 ; Microsoft, 2005).

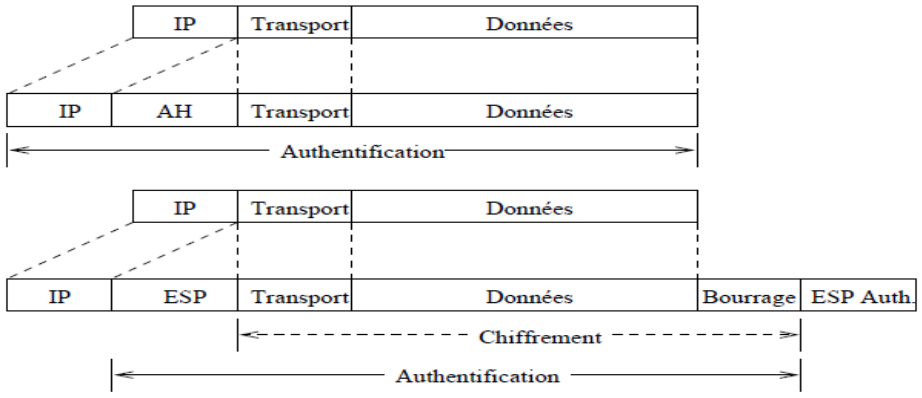

Figure 2.11 AH et ESP en mode transport (Van Quang, 2005)

2.1.3.1.2. Mode tunnel

En mode tunnel, non seulement la charge utile est protégée mais aussi l'entête IP de datagramme. Ce mode ajoute un nouvel entête au début de paquet transmis qui sert à transporter le paquet jusqu'à la fin du tunnel, où l'entête original est rétabli. Donc, le mode tunnel est utile pour sécuriser les datagrammes qui ont besoin de traverser un réseau non sécurisé. Il est utilisé dans les configurations suivantes : Passerelle à

passerelle, Serveur-passerelle, serveur à serveur. La figure 2.12 illustre les paquets IP protégés par AH-IPSec et ESP-IPSec en mode tunnel.

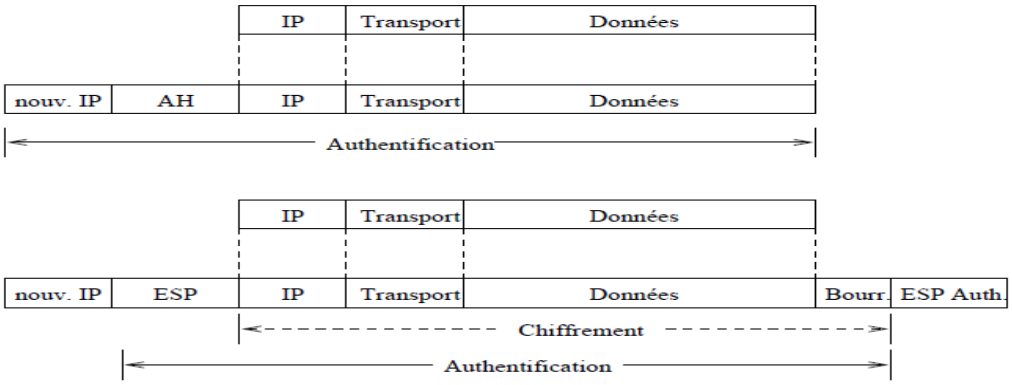

Figure 2.12 AH et ESP en mode tunnel (Van Quang, 2005)

2.1.3.1.3. Authentication Header (AH)

L'Authentication Header est utilisée pour fournir l'intégrité sans connexion, l'authentification de l'origine des données pour les datagrammes IP, et la protection contre les attaques par rejeu. Cette dernière représente un service optionnel qui peut être choisi par le récepteur quand une association de sécurité est établie. Le principe de fonctionnement de protocole AH est d'ajouter un champ supplémentaire au datagramme IP qui permet au récepteur de vérifier l'authenticité des données. Comme le montre la figure 2.13, l'AH contient les champs suivants (Kent and Atkinson, 1998a):

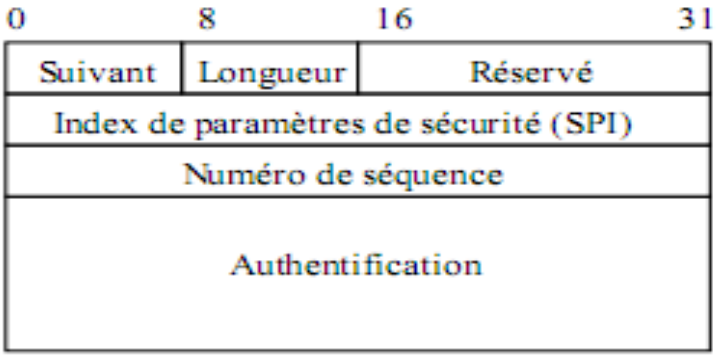

Figure 2.13 Le format AH d'IPSec (Kent and Atkinson, 1998a)

- Suivant: identification du type d'en-tête suivant cet en-tête;
- Longueur: longueur d'en-tête d'authentification;

- Réservé: usage futur;
- Index de paramètre de sécurité: identification d'une association de sécurité;
- Numéro de séquence : valeur de compteur croissant;
- Authentification : champ variable de longueur multiple de 32 bits. Ce champ contient la valeur MAC (Lasserre and Klein, 2011);

2.1.3.1.4. Encapsulating Security Payload (ESP)

ESP est utilisée pour fournir la confidentialité du contenu des communications du réseau ainsi que l'authentification du système et l'intégrité des données (Michael and Herbert, 2009). L'en-tête ESP est inséré après l'en-tête IP et avant l'en-tête du protocole de la couche supérieure (mode de transport) ou avant l'encapsulation du paquet IP (mode tunnel) (Kent and Atkinson, 1998b). Comme le montre la figure 2.14, l'ESP contient les champs suivants:

- Index des paramètres de sécurité: identification d'une association de sécurité;
- Numéro de séquence: valeur de compteur croissante;
- Données: segment de niveau transport (mode transport) ou paquet (mode tunnel) protégé par le chiffrement;
- Bourrage: remplissage des blocs à chiffrer;
- Longueur: indicateur du nombre d'octets de bourrage;
- Suivant: identification du type d'en-tête suivant cet en-tête;
- Authentification: champ variable de longueur multiple de 32 bits. Ce champ contient la valeur du MAC.

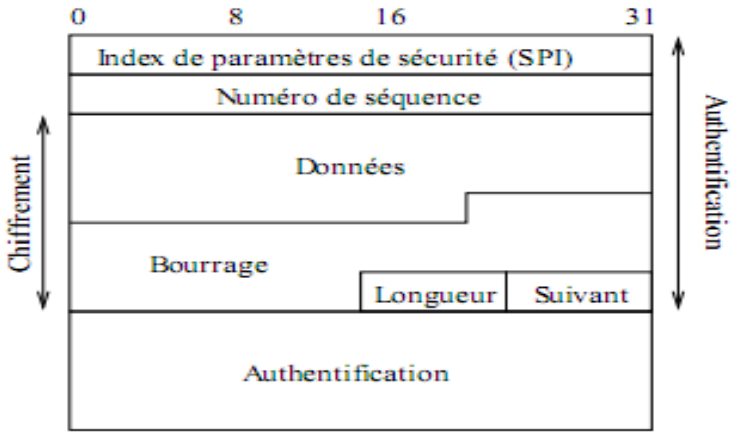

Figure 2.14 Le format d'en-tête ESP d'IPSec (Kent and Atkinson, 1998b)

Les mécanismes de sécurité AH et ESP peuvent être utilisés seuls ou combinés. Le tableau 2.1 représente les dimensions de sécurité offerts par AH et ESP.

	AH	ESP (chiffrement seul)	ESP (chiffrement & authentification)
Contrôle d'accès	Oui	Oui	Oui
Intégrité des données	Oui	Non	Oui
Non-répudiation	Oui	Non	Oui
Anti-rejeu	Oui	Oui	Oui
Confidentialité	Non	Oui	Oui
Confidentialité du flot de trafic	Non	Oui	Oui

Tableau 2.1 Les dimensions de sécurité offerts par AH et ESP (Martin, 2006)

2.1.3.2. Architecture de l'IPSec

Les protocoles IPSec comprennent AH, ESP, IKE, ISAKMP/Oakley. Pour comprendre comment mettre en œuvre et utiliser l'IPSec, il est nécessaire de comprendre la relation entre ces composants. La figure 2.15 illustre l'architecture de l'IPSec et la relation entre ces composons.

Avant que deux entités communicantes puissent échanger des communications sécurisées, elles doivent s'accorder sur la nature de la sécurité à appliquer à leurs communications: les en-têtes de sécurité (AH, ESP, ou les deux) qui seront appliqués, les algorithmes cryptographiques à utiliser, les clés secrètes…etc. Afin de gérer ces paramètres confidentiels, l'IPSec a fait recours au concept de l'association de sécurité (Security Association, SA). Une association de sécurité (SA) se compose de toutes les informations nécessaires pour faire le traitement d'IPSec sur un paquet IP (Oppliger, 2009).

Une association de sécurité (SA) est une relation à sens unique entre l'émetteur et un récepteur. Elle définit les services de sécurité pour une direction, soit entrant pour les paquets reçus par l'entité, ou sortant, pour les paquets qui sont envoyés par l'entité (Doraswamy and Harkin, 2003). Les SAs sont identifiés par trois paramètres:

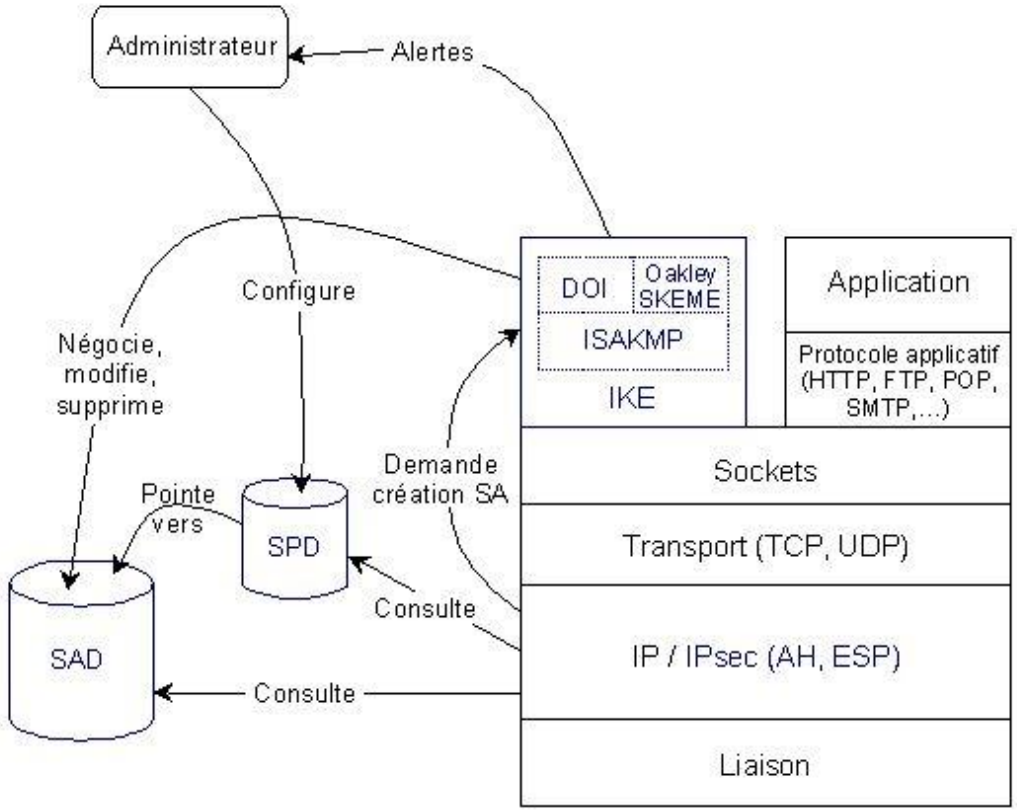

Figure 2.15 L'architecture de l'IPSec (Lasserre and Klein, 2011)

- Un index de paramètres de sécurité (SPI -Security Parameters Index): c'est un bloc de 32 bits qui circule en clair dans les en-têtes échangés. Une SPI de valeur 0 est un cas particulier pour dire qu'aucune SA n'a été encore créée;
- L'adresse de destination: elle permet de déterminer la direction de l'application de l'association de sécurité. Elle peut être un système d'extrémité ou un système intermédiaire (routeur, frewall ou poste de travail);
- L'identifiant de protocole de sécurité (SPId -Security Protocol Identifer-) indique la nature de la SA (AH ou ESP) (Labouret, 2000);

Il contient aussi d'autres paramètres ((Denizot et al., 2015; Labouret, 2000):

- Les ports source et destination (peuvent aussi jouer le rôle des paramètres pour identifier la SA);
- L'adresse IP source;
- Le nom (user ID ou un nom de système comme un nom FQDN / X.500, …);
- L'algorithme d'authentification et les clés publiques associées éventuelle;

- L'algorithme de cryptage et les clés publiques associées éventuelles;
- La durée de vie de la SA: indiquée en unité de temps ou en nombre d'octets protégés;
- Le mode d'utilisation: mode tunnel ou mode transport;
- Le numéro de séquence;
- Les paramètres utilisés pour effectuer la fragmentation de message;
- Le lien vers la SPD: c'est l'identifiant qui va permettre de trouver la correspondance dans la SPD à partir de la SAD (Oppliger, 2009; Labouret, 2000).

Typiquement, les SAs existent par paire, une dans chaque direction. Elles peuvent être créées manuellement ou dynamiquement. Dans notre thèse on s'intéresse à la création dynamique par le protocole « internet key exchange protocol ». On pratique les SAs résident dans une base de données d'association de sécurité (Security Association Database SAD). Le tableau 2.2 représente un exemple de SAD.

SPI	N.SA	IP src.	IP dest.	Port src	Port dest.	SPId	Mode	Type	N SPD	… …
150	1	10.0.0.4	Any	Any	80	ESP	Tunnel	Sortant	20	…
24	1	10.0.0.9	10.0.0.5	80	Any	AH	Transport	Entrant	14	…

Tableau 2.2 Un exemple de SAD (Martin, 2006)

Pour les paquets entrants et sortants, le SPD dicte les conditions dans lesquelles le paquet peut être accepté par l'hôte. Chaque règle se compose d'un ou plusieurs sélecteurs, les sélecteurs distinguent les actions à appliquer à ces paquets. Les sélecteurs utilisés par le SPD sont les mêmes que ceux utilisés par le SAD. Trois actions possibles peuvent être le résultat de l'application d'une règle de SPD:

- Rejeter le parquet: certains types de trafics peuvent être considérés comme inhérents d'insécurité et interdit d'être envoyés ou reçus en toute situation.

- Envoyer le paquet sans protection IPSec: un hôte ou une passerelle de sécurité permet à certains types de communications d'être envoyés ou reçus en clair.
- Appliquer la protection IPsec au paquet: si la protection de l'IPsec est nécessaire pour un paquet, le SPD indique les détails de protection: l'en-tête (s) IPSec à appliquer, les algorithmes cryptographiques qui seront utilisés, le mode d'encapsulation…etc. Un exemple de SPD est illustré dans le tableau suivant (Oppliger, 2009).

Régle	IP src.	IP dest.	Port src.	Port dest.	Action	SPId	Mode	N.SPD
1	10.0.0.2	Any	Any	23	IPSec	AH	Transport	200
2	10.0.0.1	10.0.0.3	80	Any	Drop	-	-	741
3	10.2.2.8	10.0.0.4	Any	Any	Reject	-	-	234
4	10.2.2.0	10.0.0.1	Any	Any	Accept	-	-	21

Tableau 2.3 Un exemple de SPD (Martin, 2006)

Pour mieux comprendre le fonctionnement de l'IPSec nous présentons le déroulement de trafic sortant et entrant (Caballero, 2008; Denizot et al., 2015; Lasserre and Klein, 2011) :

- Trafic sortant: lors de l'arrivée du paquet sortant à la couche IPSec, celle-ci va tout d'abord consulter la base de données de politique de sécurité afin de connaitre la politique de l'IPSec (SP) associé à ce paquet. Si le SPD indique qu'il y a une application de politique d'IPSec, alors il va consulter la base d'association de sécurité (SAD) afin de trouver l'association de sécurité qui correspond à cette politique de sécurité (SP). Dans le cas où SA existe déjà, il l'utilise directement pour traiter le paquet reçu. Autrement, l'IPSec doit appeler le protocole IKE pour négocier une SA avec le nœud destinataire du paquet. Cette partie sera détaillée dans le chapitre suivant.

- Trafic entrant: lors de l'arrivée du paquet entrant à la couche IPSec, celle-ci va tout d'abord examiner l'en-tête du paquet pour savoir si un ou plusieurs services

d'IPSec sont appliqués. Si oui, l'IPSec consultera SAD afin de déterminer la SA associée au paquet entrant. Ce dernier contient les paramètres à utiliser pour la vérification et/ou le déchiffrement du paquet. Une fois le paquet vérifié et/ou déchiffré, la SPD est consultée pour savoir si l'association de sécurité appliquée au paquet correspond bien à celle requise par les politiques de sécurité.

2.2. Conclusion

Dans ce chapitre nous avons présenté les différents protocoles de sécurité existants dans la littérature qui permettent de sécuriser les données transmisse dans les réseaux. Parmi ces protocoles, on trouve le protocole IPSec qui permet de sécuriser les données au niveau de la couche réseau, dans le chapitre suivant nous détaillons la phase d'initialisation du protocole IPSec qui utilise le protocole IKE et nous donnons les successeurs de ce protocole.

Chapitre III : Les protocoles IKEs

IPSec est une suite de protocoles IETF qui permet de sécuriser le protocole Internet (IP). En particulier, IPSec fournit la confidentialité, l'intégrité des données, le contrôle d'accès et l'authentification de la source des données. Contrairement à SSL / TLS, IPSec fournit la sécurité de bout en bout, d'une la manière transparente à l'application sans avoir à modifier chaque application séparément. La figure 3.1 illustre l'architecture du protocole IPSec. Les propriétés de sécurité d'IPSec dépendent essentiellement des protocoles d'échange des clés sous-jacents connus sous le nom du protocole IKE (Internet Key Exchange). IKE est un protocole complexe. Dans ce chapitre, une description détaillée de ce protocole est présentée, puis nous examinons les successeurs du protocole IKE afin de les comparer à nos contributions proposées dans le chapitre 5.

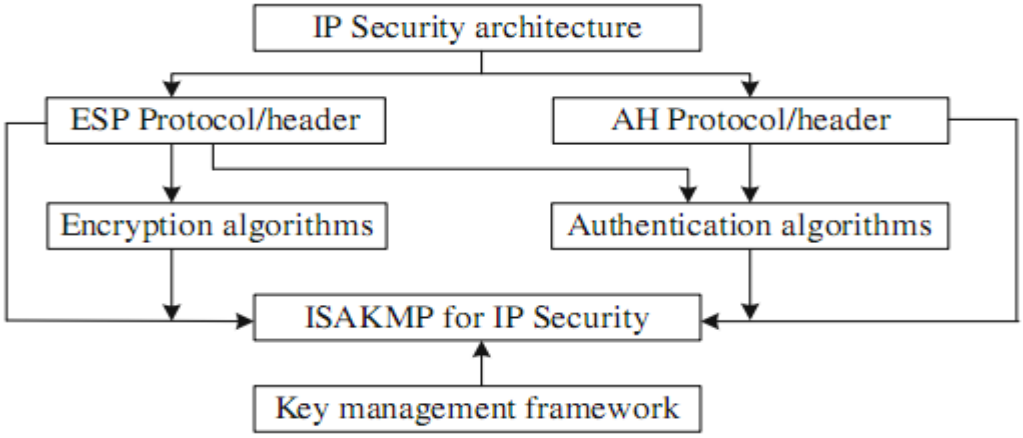

Figure 3.1 L'architecture de l'IPSec (Zheng and Zhang, 2009)

3.1 Le protocole IKE version 1

Le protocole Internet Key Exchange (IKE) est la partie principale de la mise en œuvre de l'IPSec. Il est utilisé pour négocier les clés secrètes entre les deux parties, appelées par l'initiateur et le répondeur. La clé secrète est le résultat du protocole, et il est utilisé pour créer les associations de sécurité (SA) qui définissent comment le trafic entre les deux parties doit être protégé.

IKE est spécifié par la Société Internet RFC 2409 (Request for Comments) qui fait référence au protocole ISAKMP (Internet Security Association and Key Management Protocol) et une identification de DOI (Domain of interpretation) qui

est utilisé pour interpréter le contenu des messages ISAKMP (Allard et al., 2008; Harkins and carrel, 1998 ; Perlma and Kaufman, 2001; Thomas and Elbirt, 2006; Su and Chang, 2007).

3.1.1 Fonctionnement du protocole

Le protocole IKE est conçu pour échanger les clés de chiffrement et de négocier les associations de sécurité pour la communication sécurisée. Il fonctionne en deux phases.

- La phase 1 consiste à établir une SA ISAKMP et dériver les secrets partagés qui seront utilisés pour protéger les échanges de la phase 2;
- La phase 2 consiste à négocier la SA de l'IPSec (ou d'autres services de sécurité) et génère la clé de chiffrement.

En outre, le protocole IKE définit trois modes de base d'échange : le mode principal et le mode agressif sont utilisés dans la phase 1, et le mode rapide est utilisé dans la phase 2 (Cheng, 2001; Zhu et al., 2010; Zhou, 2000).

3.1.1.1 Une association de sécurité (Security association, SA)

SA est un ensemble de politiques de sécurité utilisée pour protéger les informations. Une SA ISAKMP établie dans la phase 1 est utilisée pour protéger les messages IKE tandis qu'un SA IPSec établi dans la phase 2 est utilisé pour protéger des paquets IP.

Une SA ISAKMP comprend principalement les attributs de sécurité suivants :

- L'algorithme de hachage (hash algorithm) : indique l'algorithme de hachage qui peut être utilisé par le serveur IKE (e.g. MD5, SHA, Tiger);
- L'algorithme de chiffrement (encryption algorithm) : indique l'algorithme de chiffrement qui peut être utilisé par le serveur IKE (DES-CBC, IDEA-CBC, Blowfish-CBC, RC5-R16-B64-CBC, 3DES-CBC, CAST-CBC);
- La méthode d'authentification (authentication method): indique la méthode d'authentification qui peut être utilisée par le serveur IKE (pre-shared key, digital signature, public key encryption);
- La fonction pseudo-aléatoire (prf Algorithm): indique la fonction de pseudo-aléatoire qui peut être utilisée par le serveur IKE;

- La description du groupe pour dériver le secret partagé ;
- La durée de vie de SA.

Les attributs de sécurité à négocier dans un SA IPSec sont différents de ceux dans une SA ISAKMP (Cheng, 2001; Thomas and Elbirt, 2006).

- Le type de protocole IPSec (par exemple AH, ESP);
- L'algorithme de protocole IPSec (par exemple AH-HMAC-MD5, AH-HMAC-SHA-1, ESP-DES, ESP-3DES);
- L'algorithme d'authentification utilisé avec ESP (par exemple HMAC-MD5, HMAC-SHA-1);
- Le mode d'encapsulation (par exemple, le mode tunnel, le mode de transport);
- La description du groupe pour PFS (Perfect Forward Secrecy);
- La durée de vie SA (Cheng, 2001; Thomas and Elbirt, 2006).

3.1.1.2 Le mode principal (Main mode)

Dans le protocole mode principal, Il y a six étapes d'échange entre l'initiateur et le répondeur. Les deux premières étapes négocient les paramètres de sécurité de la SA ISAKMP; les deux secondes étapes échangent les valeurs publiques de Diffie-Hellman; et les deux dernières étapes authentifient la SA ISAKMP et l'échange de Diffie-Hellman.

RFC 2409 définit quatre méthodes d'authentification pour le protocole de mode principal: la clé pré-partagée, la signature numérique, le chiffrement à clé publique et le mode révisé de chiffrement à clé publique pour l'authentification. Nous discutons tout d'abord les caractéristiques d'IKE qui sont indépendantes de toutes les méthodes d'authentification et puis nous détaillons ces méthodes d'authentification (Cheng et al., 2001; Thomas and Elbirt, 2006).

3.1.1.2.1. Les caractéristiques communes du protocole IKE

- Les composantes publiques DH générées par l'initiateur (noté g^{Xi}) et le répondeur (noté g^{Xr}) seront mis dans les charges utiles d'échange de clé (KE).
- Le calcul des données d'authentification [AUTH] dépend de la méthode d'authentification utilisée. Mais, quelle que soit la méthode d'authentification, les

données d'authentification sont toujours calculées dans les valeurs de hachage d'information suivantes:

- $HASH_I$ = prf(SKEYID, g^{Xi} ‖ g^{Xr} ‖cookie-I‖cookie-R‖R‖SA‖ID_I). $HASH_I$ est envoyé par l'initiateur;
- $HASH_R$ = prf(SKEYID, g^{Xr} ‖ g^{Xi} ‖cookie-R‖cookie-I‖R‖SA‖ID_R). $HASH_R$ est envoyé par le répondeur;
  - Le symbole "‖" signifie la concaténation;
  - « SA » dans $HASH_I$ et $HASH_R$ est la charge utile SA envoyée par l'initiateur;
  - « Prf » est une fonction pseudo-aléatoire habituellement mise en œuvre par une clé-hash telle que HMAC; la transformation mathématique exacte est déterminée lors de la négociation des paramètres;
  - SKEYID est la clé de prf, Elle est différente pour chaque méthode d'authentification.

La sortie finale de la première phase de protocole IKE est une SA ISAKMP avec trois clés secrètes partagées exclusivement entre l'initiateur et le répondeur. Les trois clés sont:

- $SKEYID_d$ = *prf*(SKEYID, $g^{Xi\ Xr}$ ‖cookie-I‖cookie-R‖0). La clé $SKEYID_d$ est utilisée pour dériver d'autres clés dans les phases I et II du protocole IKE;
- $SKEYID_a$ = *prf*(*SKEYID*, $SKEYID_d$ ‖$g^{Xi\ Xr}$ ‖cookie-I‖cookie-R‖1). La clé $SKEYID_a$ est utilisée pour authentifier des messages de la deuxième phase de protocole IKE;
- $SKEYID_e$ = *prf*(*SKEYID*, $SKEYID_a$ ‖$g^{Xi\ Xr}$ ‖cookie-I‖cookie-R‖2). La clé $SKEYID_e$ est utilisée pour chiffrer les messages 5 et 6 en mode principal et tous les messages de la Phase II du protocole IKE;

Le secret partagé DH, $g^{Xi\ Xr}$, est la principale source d'entropie (aléatoire) pour dériver ces trois clés.

IKE définit ses propres paramètres lors de la négociation des paramètres. Ces paramètres incluent les méthodes d'authentification, les algorithmes de hachage, les algorithmes de chiffrement, les fonctions pseudo-aléatoires, et les groupes algébriques Diffie-Hellman.

Dans les sections suivantes, nous discutons les quatre méthodes d'authentification. Parmi ces méthodes, on trouve la méthode de clé pré-partagée qui représente la forme de base. Les trois autres peuvent être considérées comme des variantes de la méthode clé pré-partagée (Cheng, 2001; Thomas and Elbirt, 2006).

3.1.1.2.2. Le protocole IKE utilisant une clé pré-partagé

La figure 3.2 représente le mode principal du protocole IKE utilisant une clé pré-partagée. Une clé secrète doit être partagée exclusivement entre l'initiateur et le répondeur avant que la négociation IKE ne s'effectue. AUTH sont remplacés par $HASH_I$ et $Hash_R$. La clé SKEYID est dérivée de la clé pré-partagée (Cheng, 2001):

$$SKEYID = prf\ (preshared\text{-}key, NONCE_I \| NONCE_R)$$

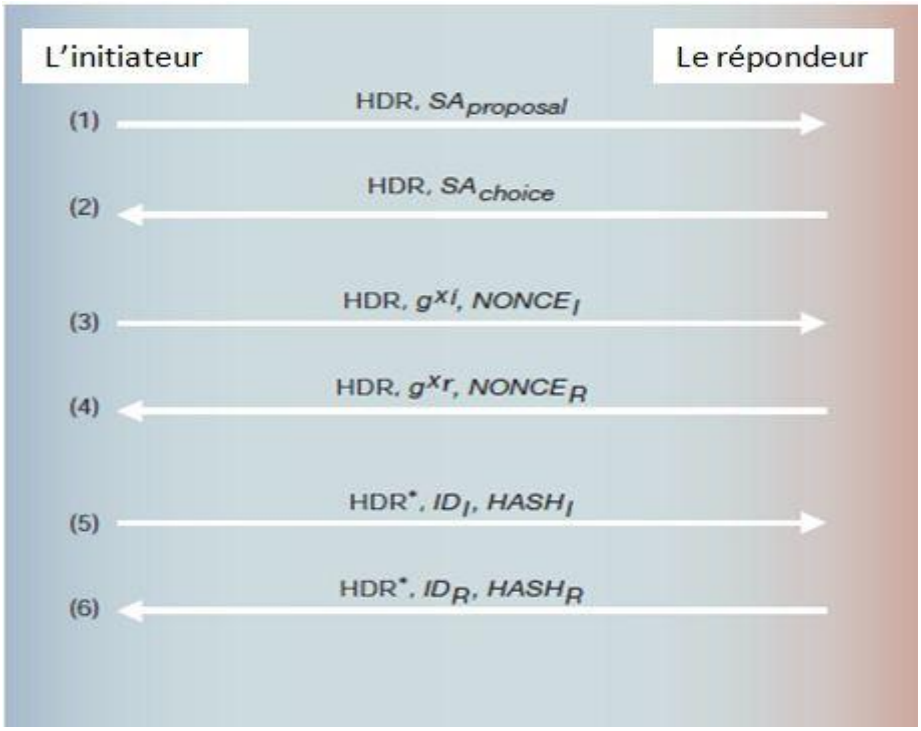

Figure 3.2 Le mode principal de protocole IKE utilisant une clé pré-partagé (Cheng, 2001)

Dans le protocole ci-dessus, HDR est un en-tête ISAKMP qui traite chaque message IKE. HDR* indique que toutes les charges utiles qui suivent le HDR sont cryptés. $SA_{proposal}$ et SA_{choice} sont les charges de l'association de sécurité. $Nonce_I$ et $Nonce_R$ sont des charges utiles de nonce. ID_I et ID_R sont des charges utiles d'identification ISAKMP. (Les indices I et R dans ces charges représentent respectivement

l'initiateur et le répondeur,). Les authentificateurs $HASH_I$ et $HASH_R$ sont générés par l'initiateur et le répondeur, respectivement (Cheng, 2001).

3.1.1.2.3. IKE utilisant la signature à clé public

La figure 3.3 illustre le mode principal du protocole IKE utilisant une signature à clé publique. AUTH sont remplacés par les signatures de l'initiateur et du répondeur, le SIG_I et le SIG_R sont calculés sur $HASH_I$ et $HASH_R$. CERTI et CERTR sont les certificats des clés publiques de l'initiateur et du répondeur. Les certificats sont placés à l'intérieur des charges utiles CERTIFICAT ISAKMP et ils peuvent être utilisés pour vérifier les signatures. L'envoi des certificats est facultatif. Si aucun certificat n'est envoyé, alors les deux parties doivent acquérir un autre certificat à travers d'autre canal, habituellement un PKIX. Les certificats doivent être vérifiés avant d'être utilisés pour vérifier les signatures. La clé SKEYID est dérivée à partir des nonces comme suit:

$$SKEYID = prf (NONCE_I \| NONCER, g^{XiXr})$$

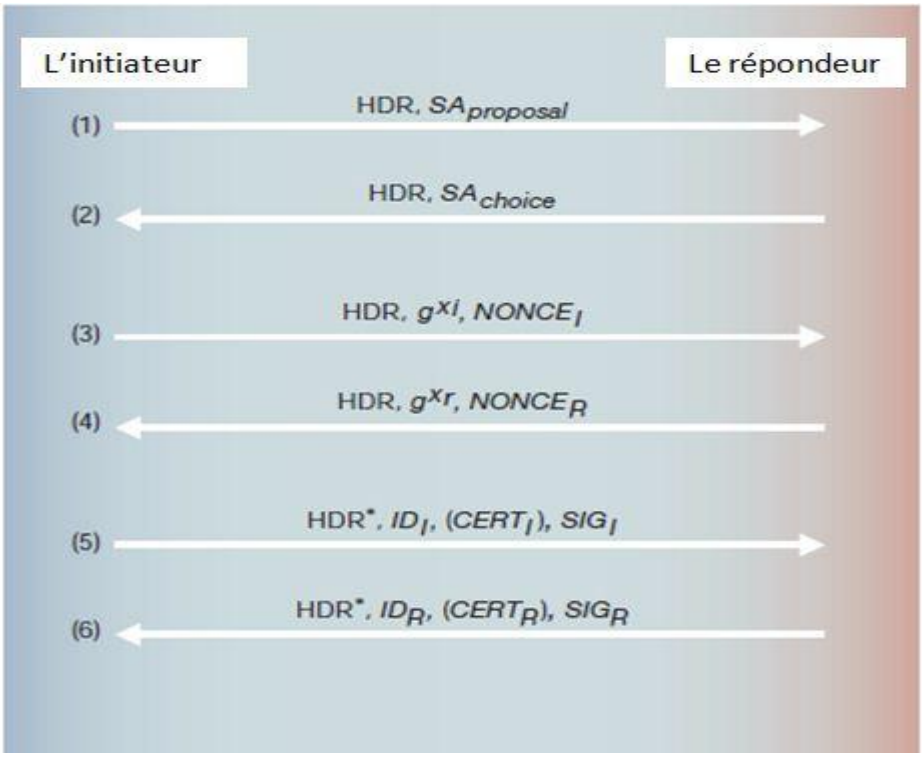

Figure 3.3 IKE utilisant une signature à clé public (Cheng, 2001)

La clé SKEYID est fraiche parce qu'elle dépend des paramètres générés aléatoirement, mais cette clé ne peut être utilisée pour l'authentification, car ni les

nonces ni le secret de DH, g^{XiXr}, sont cryptographiquement liés à l'identité de l'initiateur ou du répondeur. L'authentification est assurée par les signatures à clé publique (Cheng, 2001).

3.1.1.2.4. IKE utilisant le cryptage à clé publique

Figure 3.4 représente le mode principal du protocole IKE utilisant le cryptage à clé publique. PK_I et PK_R sont les clés publiques de l'initiateur et du répondeur. Notons que les nonces sont chiffrés avec la clé publique du destinataire prévu. Puisque seul le titulaire de la clé privée correspondante peut déchiffrer un nonce chiffré, les nonces deviennent des secrètes partagées entre l'initiateur et le répondeur. L'idée est d'utiliser les deux nonces pour remplacer une clé pré-partagée avec un avantage supplémentaire que les nonces sont éphémères et non des secrets partagés à long terme. La clé SKEYID est dérivée à partir des nonces:

$$SKEYID = prf\ (hash\ (NONCE_I \| NONCE_R),\ cookie\text{-}I \| cookie\text{-}R)$$

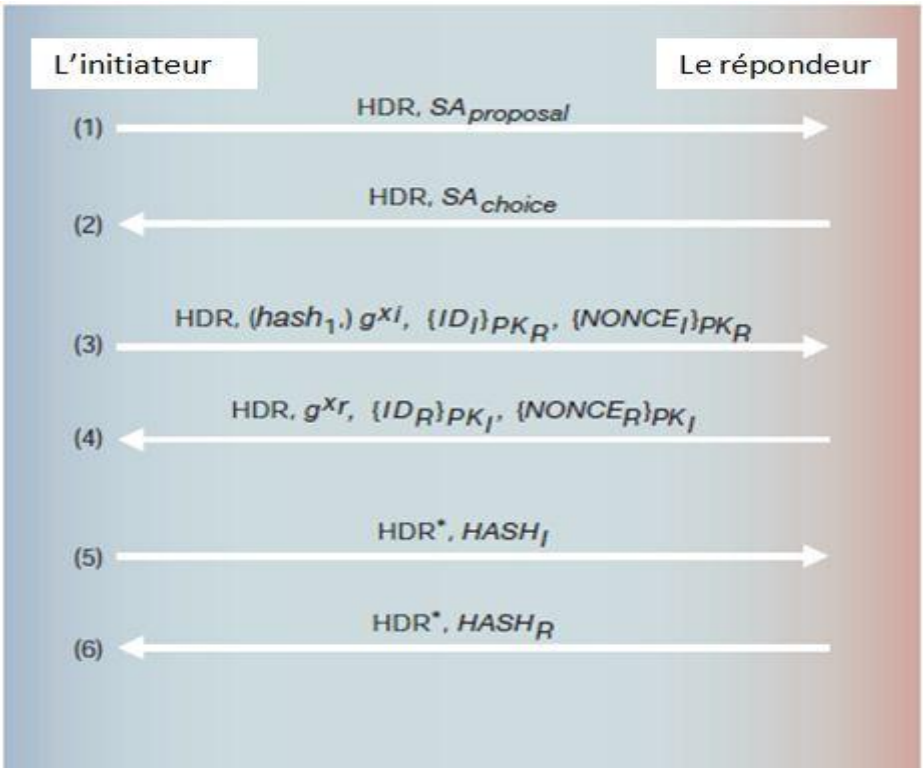

Figure 3.4 IKE utilisant le cryptage à clé publique (Cheng, 2001)

Où "hash" est un algorithme de hachage déterminé lors de la négociation des paramètres. Les nonces sont des secrets partagés, la clé SKEYID est aussi un secret

partagé; $HASH_I$ et $HASH_R$ peuvent être utilisés directement pour l'authentification (Cheng, 2001).

3.1.1.2.5. IKE utilisant le chiffrement à clé publique révisée

La figure 3.5 représente le mode principal du protocole IKE utilisant le cryptage à clé publique révisée. La clé SKEYID est le même que celui d'IKE utilisant le cryptage à clé publique. IKE utilisant le chiffrement à clé publique révisée corrige deux défauts d'IKE, utilisant le cryptage à clé publique.

Premièrement, le nombre total d'opérations de chiffrement à clé publique est réduit de quatre à deux. Les nonces sont toujours cryptés avec les clés publiques. Deux clés de chiffrement symétrique, Ke_I et Ke_R, sont dérivées, elles sont utilisées pour crypter les paramètres d'échange entre l'initiateur et le répondeur. L'algorithme de chiffrement à clé symétrique est déterminé au cours de la négociation des paramètres (Cheng, 2001).

$$Ke_I = prf\ (NONCEI, cookie\text{-}I)$$

$$Ke_R = prf\ (NONCER, cookie\text{-}R)$$

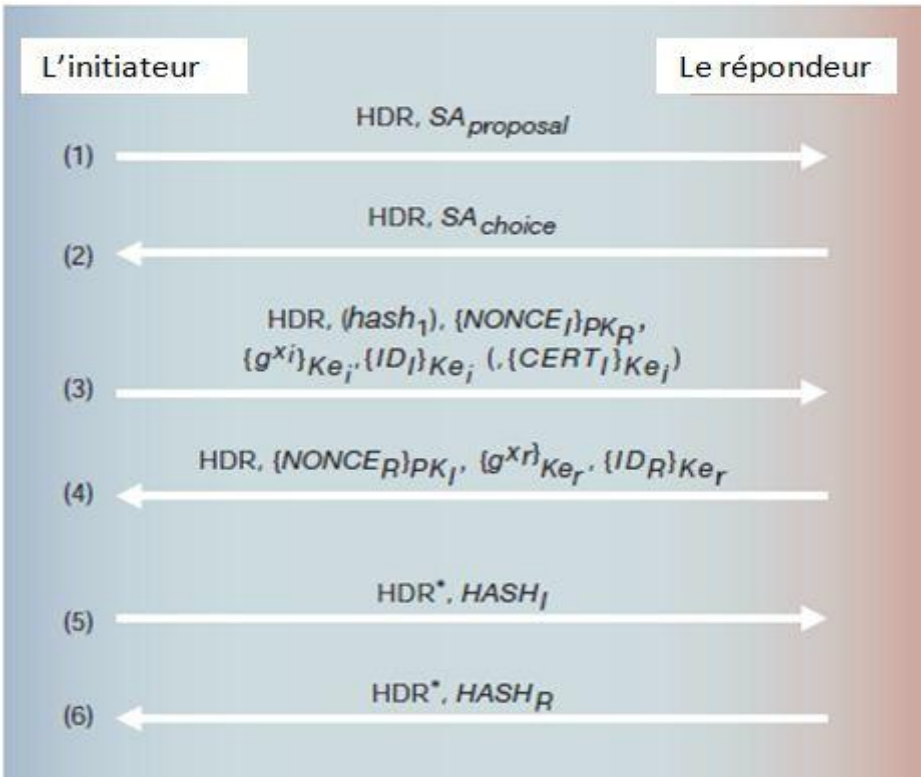

Figure 3.5 IKE utilisant le chiffrement à clé publique révisée (Cheng, 2001)

Deuxièmement, l'initiateur est autorisé à envoyer son certificat de clé publique cryptée au répondeur de sorte que celui-ci peut utiliser la clé publique à l'intérieur du certificat pour crypter $NONCE_R$. Le certificat peut être chiffré vu que le cryptage à clé symétrique est utilisé afin que la taille du certificat ne pose pas de problème.

3.1.1.3 Le mode agressif

La figure 3.6 représente le mode agressif du protocole IKE utilisant les différentes méthodes d'authentification. Les caractéristiques de ce mode sont :

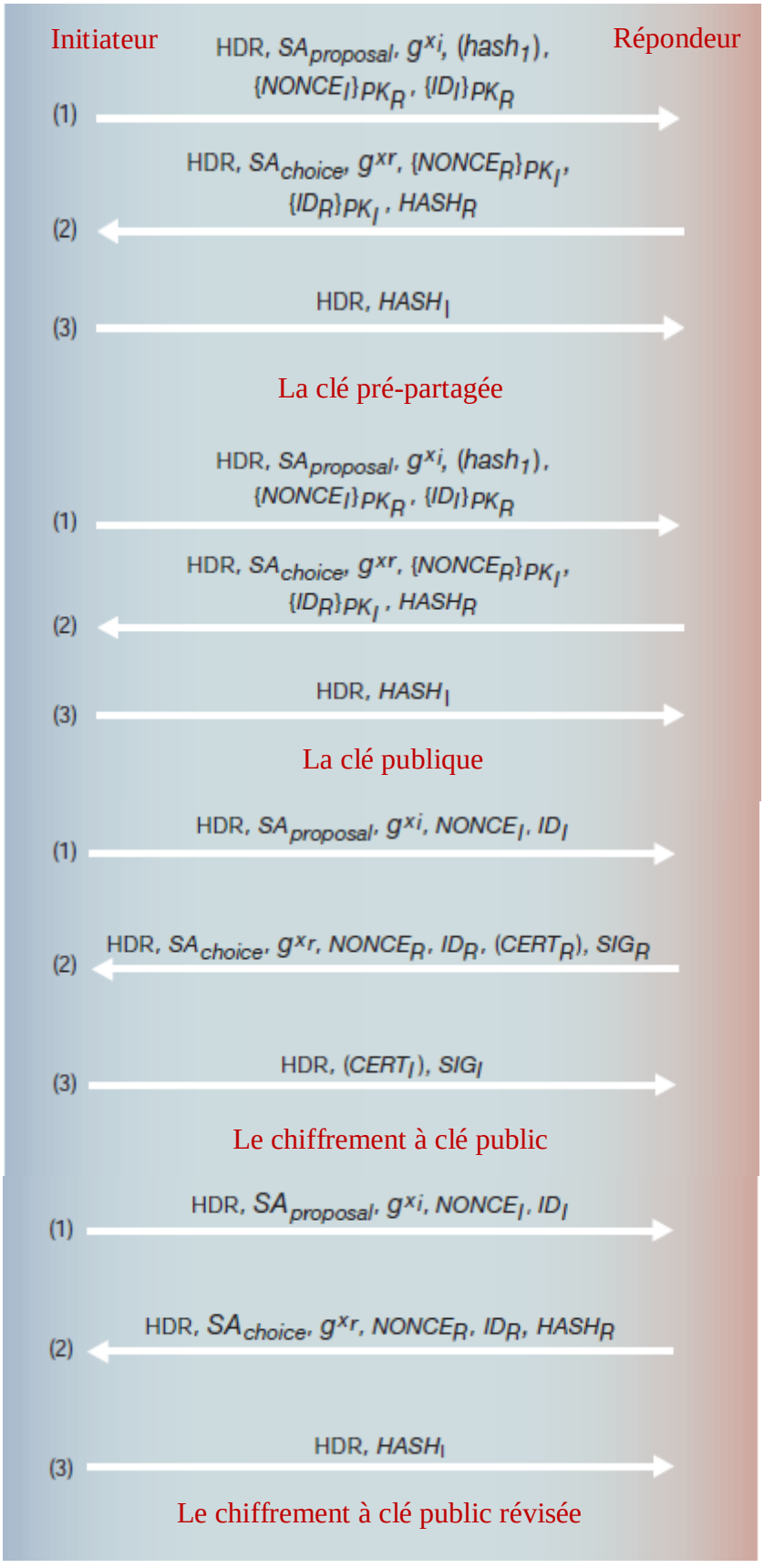

Figure 3.6 Le protocole IKE en mode agressif (Cheng, 2001)

- Les identités ne sont pas cryptées, sauf dans la méthode de chiffrement à clé publique ou révisée;
- Le groupe algébrique Diffie-Hellman ne peut être négocié. Il est choisi par l'initiateur;
- La méthode d'authentification ne peut être négociée dans le cas où l'initiateur choisit la méthode de chiffrement à clé publique ou le chiffrement à clé publique révisée. Mais, il peut offrir au répondeur un choix lors de l'utilisation de méthode à clé pré-partagée ou une signature à clé publique (Cheng, 2001).

3.1.1.4 Le mode rapide

Comme son nom l'indique, le mode rapide est proposé pour être rapide et efficace. Une négociation de mode rapide se compose de trois messages et il peut se conduire sans l'utilisation de toutes les opérations de cryptographie à clé publique.

Le mode rapide fournit la possibilité d'utiliser des techniques D-H, de sorte que g^{Xi} et g^{Xr} illustrés dans la figure suivante soient facultatifs. En outre, les identités ID_{Ui} et ID_{Ur} sont facultatives; si les identités ne sont pas envoyées, il suppose que les identités non envoyées sont les mêmes que l'ID_I et ID_R dans le SA ISAKMP. De plus, si l'initiateur envoie g^{Xi} ou (ID_{Ui} , ID_{Ur}) dans le premier message, le répondeur doit envoyer g^{Xr} (ID_{Ui} et ID_{Ur}) dans le second message pour continuer la négociation.

Dans la figure 3.7, le symbole « * » après les en-têtes des messages indiquent que les corps de ces messages sont cryptés par la clé $SKEYID_e$ et l'algorithme de chiffrement utilisé se trouve dans SA ISAKMP. $HASH_1$, $HASH_2$ et $HASH_3$ authentifient les messages correspondants. Ils sont calculés par la clé $SKEYID_a$ et la fonction pseudo-aléatoire dans la SA ISAKMP:

$$HASH_1 = prf\ (SKEYID_a, message\text{-}ID \| SA \| NONCE_I\ [\| g^{Xi}]\ \ [\| ID_{Ui} \| ID_{Ur}])$$

$$HASH_2 = prf\ (SKEYID_a, message\text{-}ID \| SA \| NONCE_R\ [\| g^{Xr}]\ \ [\| ID_{Ui} \| ID_{Ur}])$$

$$HASH_3 = prf(SKEYID_a, 0 \| message\text{-}ID \| NONCE_I \| NONCE_R)$$

Les informations entre crochets ([]) sont facultatives lors du calcul des valeurs de hachage; ils sont utilisés si et seulement si les messages contiennent ses informations.

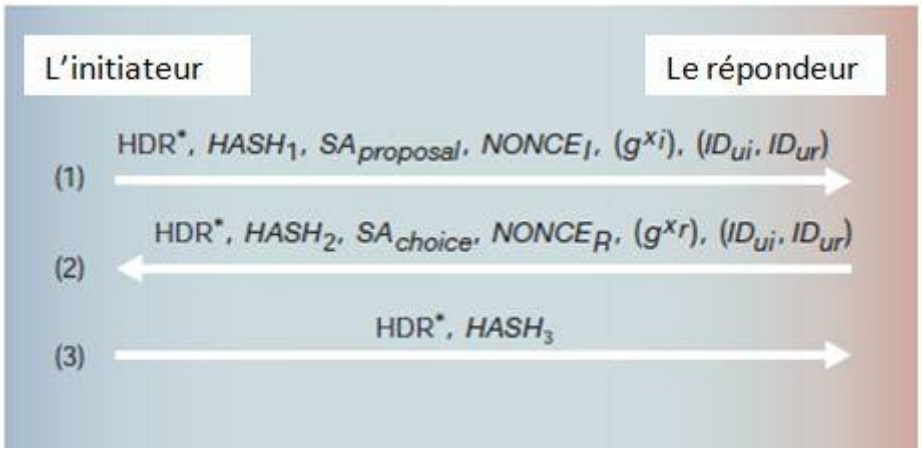

Figure 3.7 Le mode rapide du protocole IKE (Cheng, 2001)

Les propositions contenues dans la charge utile SA sont pour la sécurité du protocole IPSec. Les paramètres de sécurité de l'IPSec sont générés pour chaque protocole de sécurité et ils sont choisis par le répondeur. La clé du protocole, appelé KEYMAT, est dérivée comme suit (Cheng, 2001):

$$KEYMAT = prf(SKEYID_d, [g^{XiXr}]protocol\|SPI\|NONCE_I\|NONCE_R)$$

3.2 Le protocole IKE version 2

IKE est le protocole utilisé pour établir une association de sécurité (SA) dans la suite du protocole IPSec. IKEv2 est la seconde version du protocole IKE. L'IETF a délivré IKEv2 (la nouvelle version de protocole IKE) en décembre 2005 dans RFC 4306 (Kaufman, 2005; Kaufman et al., 2010).

3.2.1 Le fonctionnement d'IKEv2

IKEv2 conserve le cadre du protocole IKEv1 et son processus d'échange des clés est également divisé en deux étapes (Haddad and Mirmohamadi, 2005; Lu et al., 2008):

- L'étape 1: Echange initiale (The initial exchange): est utilisé pour construire la SA IKE (IKE Security Association) et la SA CHILD (Security Association Enfant) entre les entités communicantes.
- L'étape 2: CREATE-CHILD-SA (builds Child Security Association) de l'échange: est utilisé lorsqu'on a besoin de plus de SA CHILD pour protéger les données d'application ou une nouvelle SA IKE.

IKEv2 utilise les paramètres suivants:

- HDR: l'en-tête de message;
- SA_x: l'association de sécurité envoyée par x (l'initiateur ou répondeur);
- KE_x: les valeurs publiques de Diffie-Hellman (DH) envoyées par x (l'initiateur ou répondeur);
- N_x : nombres pseudo-aléatoires envoyés par x;
- SK {données}: fournit le chiffrement et l'intégrité des données;
- ID: les informations d'identité;
- AUTH: les informations d'authentification de l'identité;
- CERT: certificat;
- CERTREQ: certificat des informations de demande;
- [CERT]: est une charge utile de CERT et elle est facultative.

3.2.1.1 L'échange initiale « initial exchange »

Le premier échange dans l'IKEv2 correspond à la première étape dans IKEv1; son processus d'échange est effectué par quatre messages, comme il est représenté dans la figure suivante.

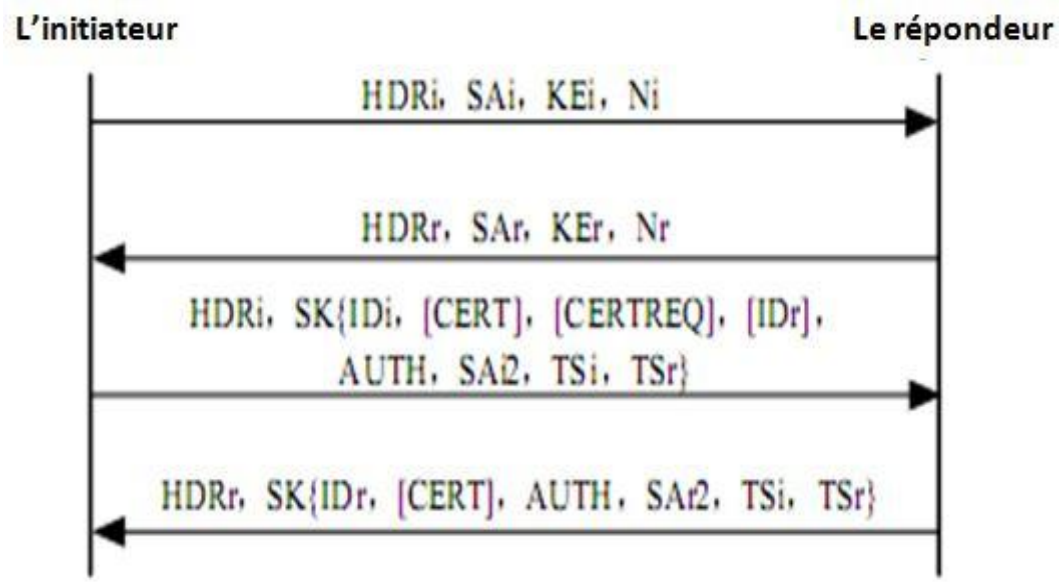

Figure 3. 8 L'échange initiale « initial exchange» (Lu et al., 2008)

Dans le premier et le deuxième message, l'initiateur et le répondeur négocient les paramètres confidentiels (SA_i et SA_r), ils échangent les valeurs publiques de D-H (KE_i et KE_r) et les nombres pseudo-aléatoires (Ni, Nr). Après la négociation, les

deux entités communicantes génèrent leurs propres clés, une clé de chiffrement SK_{ei} pour l'initiateur et SK_{er} pour le répondeur ainsi qu'une clé d'authentification SK_{ai} pour l'initiateur et SK_{ar} pour le répondeur.

Dans le troisième et quantième message, les deux entités communicantes échangent AUTH mutuellement, et ils négocient les paramètres de la SA CHILD simultanément. Si l'authentification de l'identité est passée, la SA IKE et la SA CHILD pourraient être construites entre les entités.

3.2.1.2 L'échange CREATE-CHILD-SA

L'échange CREATE-CHILD-SA correspond à la deuxième étape dans IKEv1 qui est utilisé pour construire un SA CHILD sous la protection de la SA IKE. L'échange CREATE-CHILD-SA est utilisé dans SA IKE et la nouvelle négociation SA CHILD. Le processus d'échange est composé uniquement de deux messages, comme le représente la Figure 3.9.

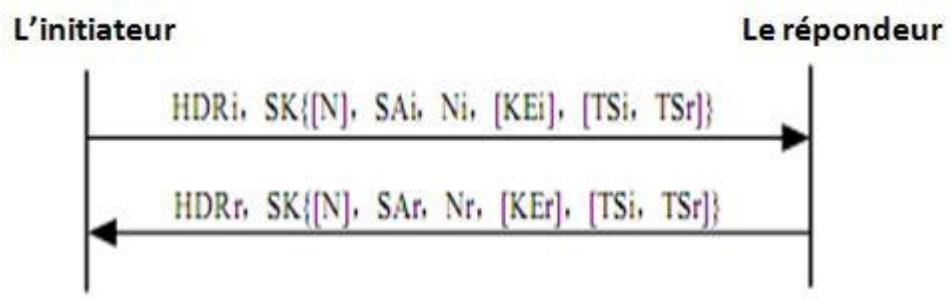

Figure 3.9 L'échange CREATE-CHILD-SA (Lu et al., 2008)

Dans cette étape, les deux entités de communication doivent négocier les caractéristiques de cryptologie de SA CHILD. Les charges utiles (SA_i et SA_r) sont utilisées pour négocier les algorithmes de la SA CHILD. Si les deux entités communicantes veulent vérifier la propriété « perfect forward protection » de la SA CHILD, elles peuvent utiliser KE_i, KE_r pour effectuer un nouvel échange Diffie-Hellman.

3.2.1.3 L'échange de l'information

Les deux parties de la communication ont besoin de transmettre des messages de contrôle pendant la période de négociation des clés. Ces messages ont pour but d'avertir des erreurs ou des remarques des points importants de l'autre partie, et ces

travaux peuvent être complétés par l'échange d'informations. Le processus d'échange d'information est représenté dans la figure 3.10.

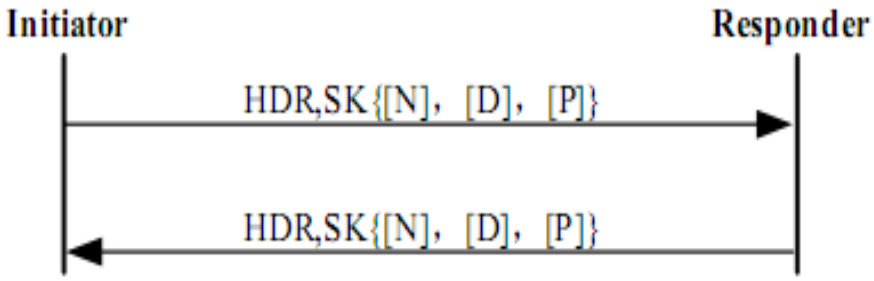

Figure 3.10 L'échange de l'information (Lu et al., 2008)

L'échange d'informations peut contenir plusieurs types d'information telle que :

- Les charges utiles de notification (N);
- Les charges utiles de suppression (D);
- Les charges utiles de configuration (P) ou sans charge utile, par exemple, une partie de communication peut transmettre un message sans charge utile d'échange d'informations pour détecter si l'autre partie est encore dans un état actif.

3.3 Le protocole JFK (Aiello et al., 2002)

Un autre successeur du protocole IKE est Juste Fast Keying. Il existe deux variantes de ce protocole. Les deux variantes notées JFKi et JFKr sont très similaires dans de nombreux aspects, avec deux différences principales: JFKi fournit une protection de l'identité active pour l'initiateur et aucune protection de l'identité pour le répondeur. Ce type de protection est approprié pour un scénario client-serveur où l'initiateur (le client) protège son identité et l'identité du répondeur est publique. Alors que JFKr fournit une protection de l'identité active pour le répondeur et une protection de l'identité passive pour l'initiateur. Ce type de protection est approprié pour un scénario peer-to-peer où le répondeur souhaite protéger son identité. La figure 3.11 représente le protocole JFK_i. Ce protocole utilise la conception de base du protocole ISO 9798-3 d'échange de clés (Aiello et al., 2002).

$$I \rightarrow R \quad : \quad N_I, g^i, ID_{R'} \tag{1}$$

$$R \rightarrow I \quad : \quad N_I, N_R, g^r, \text{grpinfo}_R, \text{ID}_R, S_R[g^r, \text{grpinfo}_R], H_{\text{HK}_R}(g^r, N_R, N_I, IP_I) \tag{2}$$

$$I \rightarrow R \quad : \quad N_I, N_R, g^i, g^r, H_{\text{HK}_R}(g^r, N_R, N_I, IP_I), \{\text{ID}_I, \text{sa}, S_I[N_I, N_R, g^i, g^r, \text{ID}_R, \text{sa}]\}_{K_a}^{K_e} \tag{3}$$

$$R \rightarrow I \quad : \quad \{S_R[N_I, N_R, g^i, g^r, \text{ID}_I, \text{sa}, \text{sa}'], \text{sa}'\}_{K_a}^{K_e} \tag{4}$$

Figure 3.11 Le protocole JFKi (Aiello et al., 2002)

JFK_i nécessite seulement quatre messages. JFK_i est optimisée pour protéger le répondeur contre les attaques DOS. Les paramètres utilisés dans ce protocole sont:

- IP_i : l'adresse réseau de l'initiateur ;
- g^x : exponentielles Diffie-Hellman (DH) ;
- g^i : exponentielle courante (mod p) de l'initiateur;
- g^r : exponentielle courante (mod p) du répondeur;
- N_I : nonce de l'initiateur, une chaine de bits aléatoires;
- N_R : nonce du répondeur, une chaine de bits aléatoires;
- ID_I : certificat de l'initiateur ou des informations d'identification à clé publique;
- ID_R : certificat du répondeur ou informations d'identification à clé publique; $ID_{R'}$: une indication envoyée par l'initiateur au répondeur pour indique les informations d'authentification qui devrait être utilisées (par exemple, certificats);
- $H_k(M)$: c'est le condensé du message M (e.g. HMAC) par la clé K telle que H est une fonction pseudo-aléatoire;
- $\{M\}^{Ke}_{Ka}$: c'est le chiffrement de message M par une clé symétrique K_e, suivi par l'authentification MAC avec clé symétrique K_a de ce message;
- S_X [M]: c'est la signature numérique du message M avec la clé privée X.

3.4 Le protocole proposé par (Haddad et al., 2004)

La figure 3.12 illustre le protocole proposé pour la première phase du protocole IKE. Ce protocole est une version modifiée du protocole d'authentification de Diffie-Hellman. Dans ce protocole chaque partie peut calculer la clé partagée. La principale caractéristique de ce protocole est la résistance contre les attaques DoS.

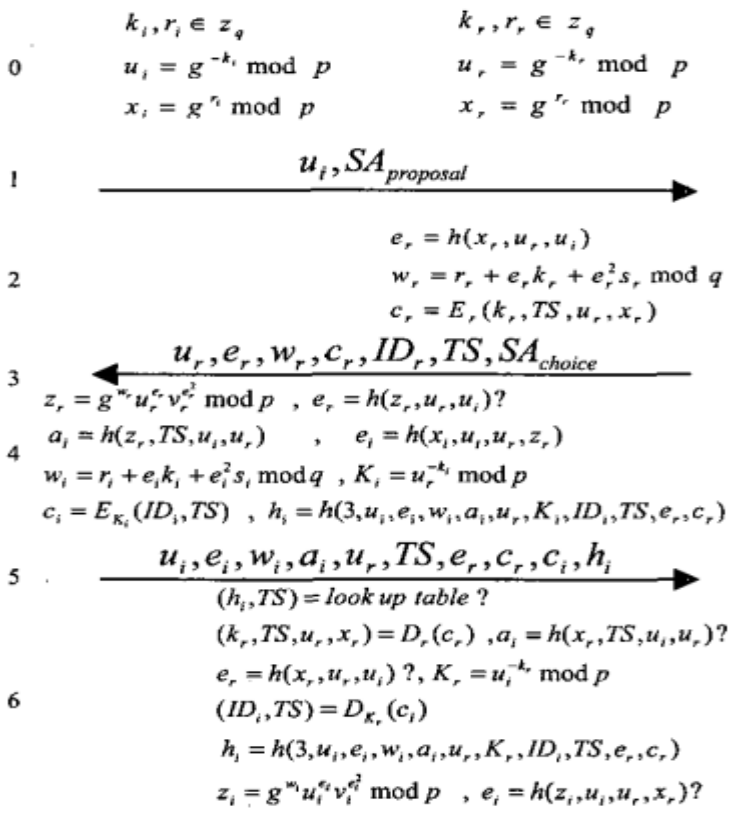

Figure 3.12 La première phase du protocole IKE (Haddad et al., 2004)

3.5 Le protocole IKE proposé par (Yang Su et al., 2007)

Yang Su et al. ont proposé un nouveau protocole Internet Key Exchange, qui a été inspiré du protocole de Haddad. Il y a au total quatre échanges de messages (deux allers-retours) dans ce protocole. Les auteurs de ce protocole ont montré que le protocole proposé et plus efficace à la résistance contre l'attaque par rapport au protocole proposé par Haddad et al. La figure 3.13 présente la version d'IKE proposée par Yang Su et al.

Comme le montre la figure 3.13 les deux entités communicantes ont chacune une paire de clés public/privée à long terme (s_i, v_i) pour l'initiateur et (s_r, v_r) pour le répondeur, et deux paires de clés éphémères (k_i, u_i) et (r_i, x_i) pour l'initiateur, et (k_r, u_r) et (r_r, x_r) pour le répondeur.

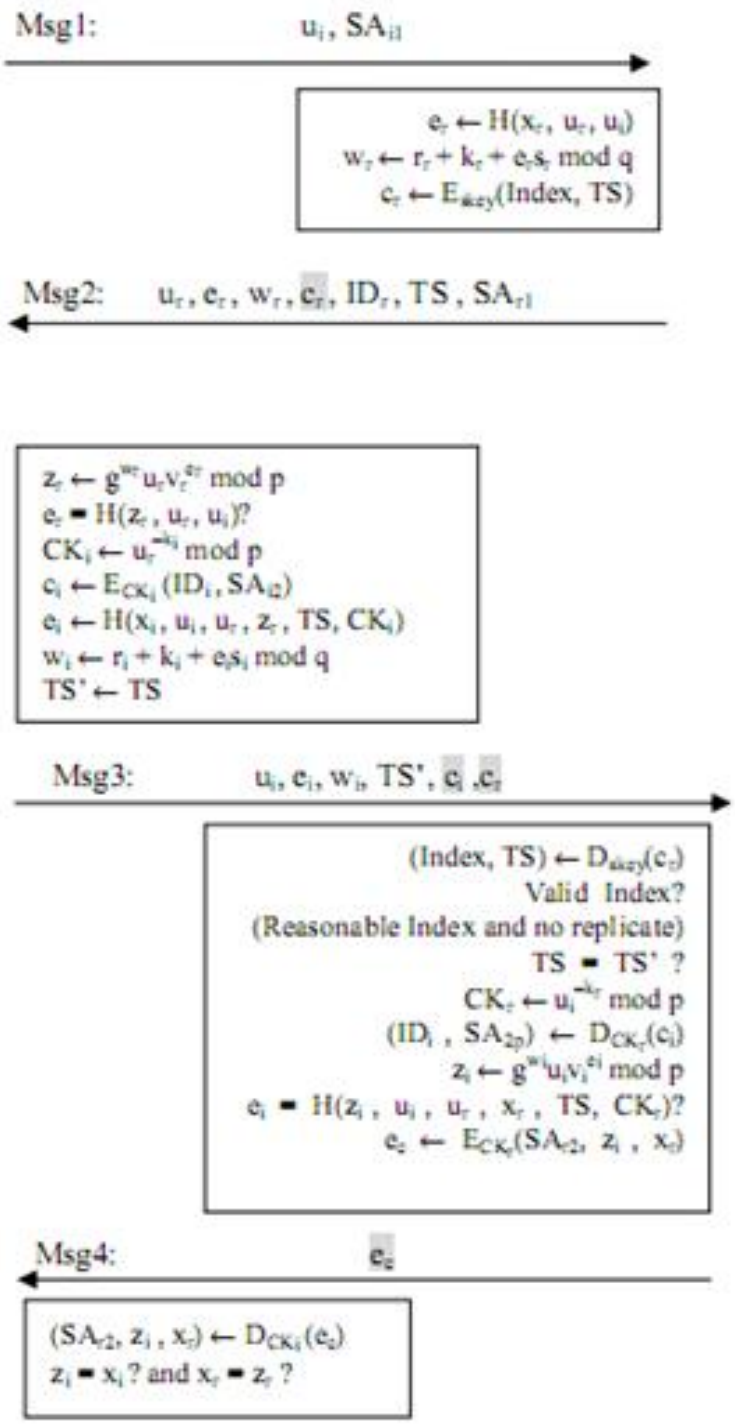

Figure 3.13 Le protocole IKE proposé par (Yang Su et al., 2007)

3.6 Le protocole IKE proposé par (Ray et al., 2012)

Ce protocole possède deux phases. La première phase consiste à établir l'association de sécurité du protocole IKE qui se compose de quatre messages, et la deuxième phase consiste à établir l'association de sécurité du protocole IPSec qui se compose de trois messages. Ce protocole utilise les courbes elliptiques Diffie-Hellman au lieu RSA basé sur Diffie-Hellman.

- Les paramètres utilisés et calculés soit par l'initiateur ou répondeur de ce protocole sont :
  - p, n: deux grands nombres premiers;
  - Fp: un corps fini;
  - E: une courbe elliptique définie sur un corps fini Fp;
  - G: le groupe de courbe elliptique points E;
  - P: un point sur une courbe elliptique E à l'ordre n;
  - I: l'initiateur;
  - R: le répondeur;
  - HDR: ISAKMP-tête;
  - SA_{OFFD} : une liste de propositions cryptographiques de l'initiateur ;
  - SA_{SELEC} : les protocoles cryptographiques sélectionnés par le répondeur de la liste envoyée par l'initiateur;
  - ID_I: l'identité de l'initiateur I;
  - ID_R: l'identité du répondeur R;
  - N_I: Nonce de l'initiateur;
  - N_R: Nonce du répondeur ;
  - (k_i, PU_I): une paire de clé privée- publique de l'initiateur où la clé publique $PU_I = k_i.P$;
  - (k_r, PU_R): une paire de clé privée- publique du répondeur R, où $PU_R = k_R.P$;
  - $X_I = prf (IP_I \mid ID_I \mid PU_I)$;
  - $X_R = prf (IP_R \mid ID_R \mid PU_R)$;
  - $SKEYID = prf (Skey\_e , N_I \mid N_R)$;
  - $HASH\text{-}I = prf (SKEYID, IP_I, IP_R \mid SA_{OFFD} \mid ID_I)$;
  - $HASH\text{-}R = prf (SKEYID, IP_I, IP_R \mid SA_{OFFD} \mid ID_R)$;
  - $HASH\ 1 = prf (K_Y, MsgID \mid SA \mid N_I)$;

- HASH 2 = prf (K_Y, MsgID| SA| N_R);
- HASH 3 = prf (K_Y, MsgID| SA| N_I| N_R).

❖ Phase-I du protocole proposé: dans ce protocole, l'initiateur doit connaitre l'identité et l'adresse IP du répondeur souhaitée pour lancer la négociation de clé. L'adresse IP du répondeur est unique et son système est protégé par mot de passe. Les détails de la phase I du protocole sont représentés dans la figure 3.14.

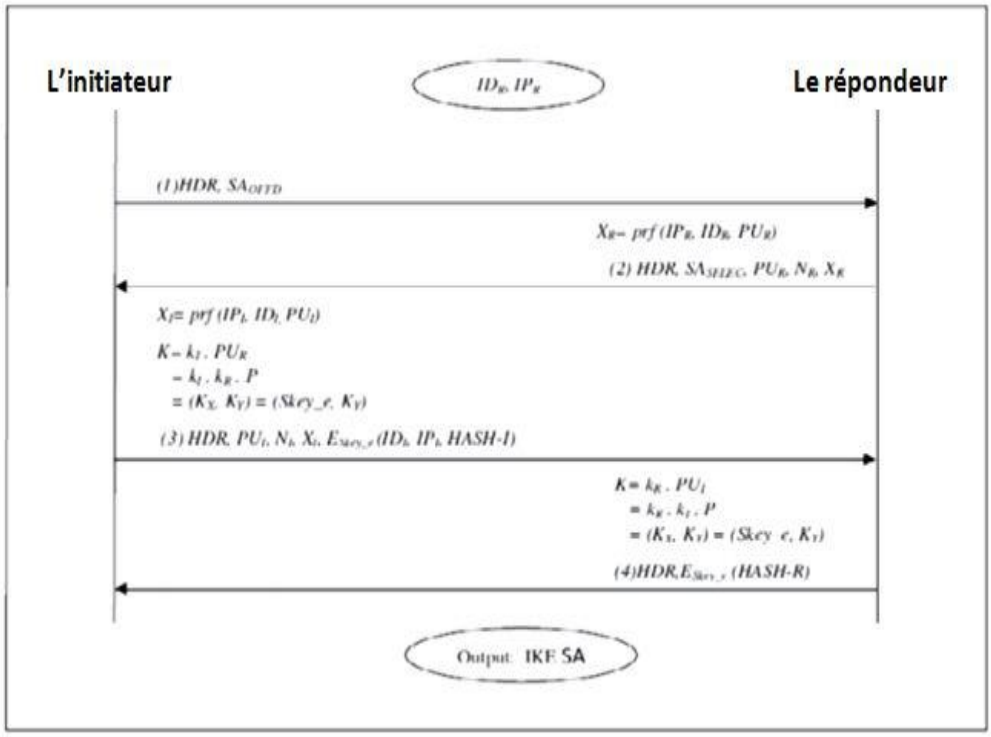

Figure 3.14 La phase I du protocole proposé par (Ray et al., 2012)

❖ Phase-II du protocole proposé: Après l'achèvement réussi de la phase I, le mode rapide utilise la SA IKE pour créer la SA IPSec. Le SA IKE protège l'échange de mode rapide par cryptage et authentification des messages. Les détails de la phase II du protocole IKE sont présentés dans la figure 3.15.

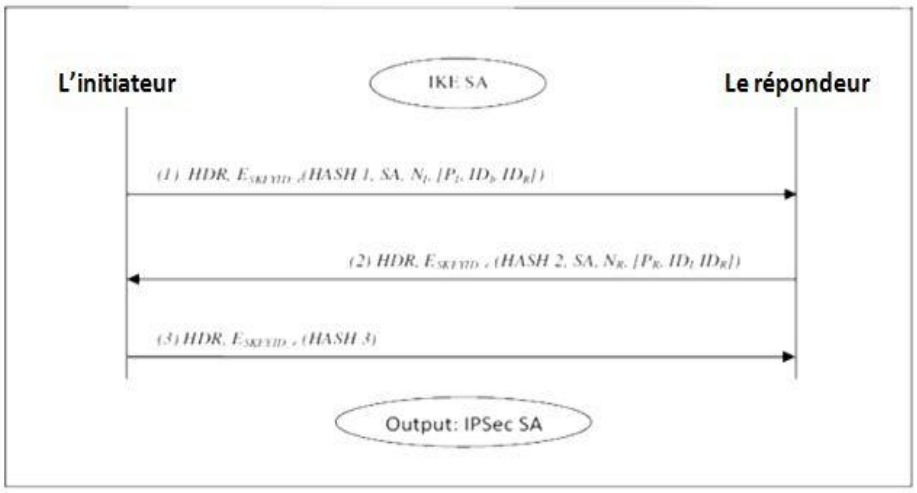

Figure 3.15 La phase II du protocole proposé par (Ray et al., 2012)

3.7 Les modifications proposées par (Nagalakshmi et al., 2011) du protocole IKE

(Nagalakshmi et al., 2011) ont proposé des modifications de la première et la deuxième phase du protocole IKE en mode principale et rapide pour les différentes méthodes d'authentification.

3.7.1 Modification de la première phase du protocole IKE en mode principal basé sur la signature à clé public (Nagalakshmi et al., 2011)

Dans ce mode, les deux parties communicantes ont des clés publiques capables de faire des signatures. La figure suivante illustre la proposition de Nagalakshmi et al. pour la phase 1.

Comme le montre la figure 3.16, les messages 3 et 4 comprennent la valeur de hachage de la clé privée de l'expéditeur et les valeurs privées Diffie-Hellman. Les messages 5 et 6 sont utilisés pour l'authentification.

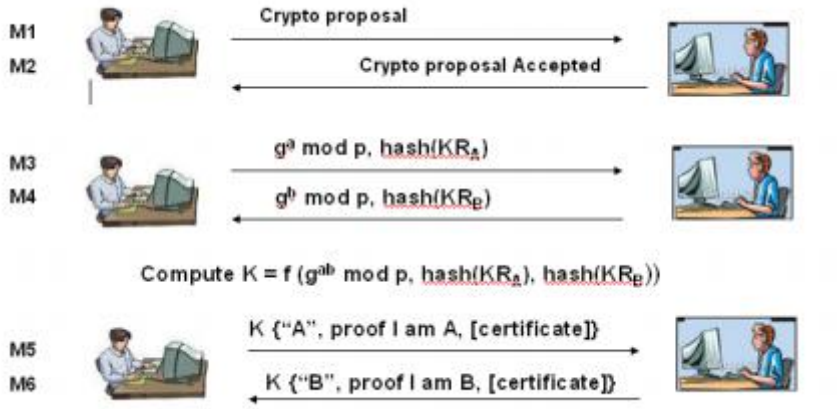

Figure 3.16 La modification du protocole IKE basé sur la signature à clé publique (Nagalakshmi et al., 2011)

3.7.2 La modification de la première phase du protocole IKE en mode principal basé sur le chiffrement à clé public (Nagalakshmi et al., 2011)

La figure 3.17 illustre ce protocole. Dans ce protocole, dans l'étape 3, l'initiateur crypte les informations d'authentification et son identité avec la clé publique du répondeur. À son tour, le répondeur crypte les informations d'authentification et son identité avec la clé publique de l'initiateur. Ensuite, la clé secrète partagée est calculée. L'étape 5 et 6 consistent à authentifier l'initiateur et le répondeur.

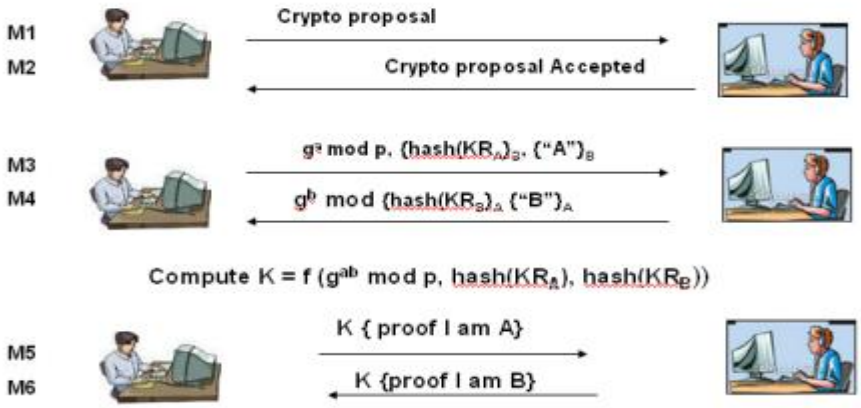

Figure 3.17 La modification de la première phase du protocole IKE en mode principal basé sur le chiffrement à clé public (Nagalakshmi et al., 2011)

3.7.3 La modification de la deuxième phase du protocole IKE en mode rapide

Ce protocole est composé de trois messages qui négocient les paramètres de l'association de sécurité de l'IPSec. Tous les messages de ce protocole sont cryptés par la clé qui est créée à la fin de la première phase.

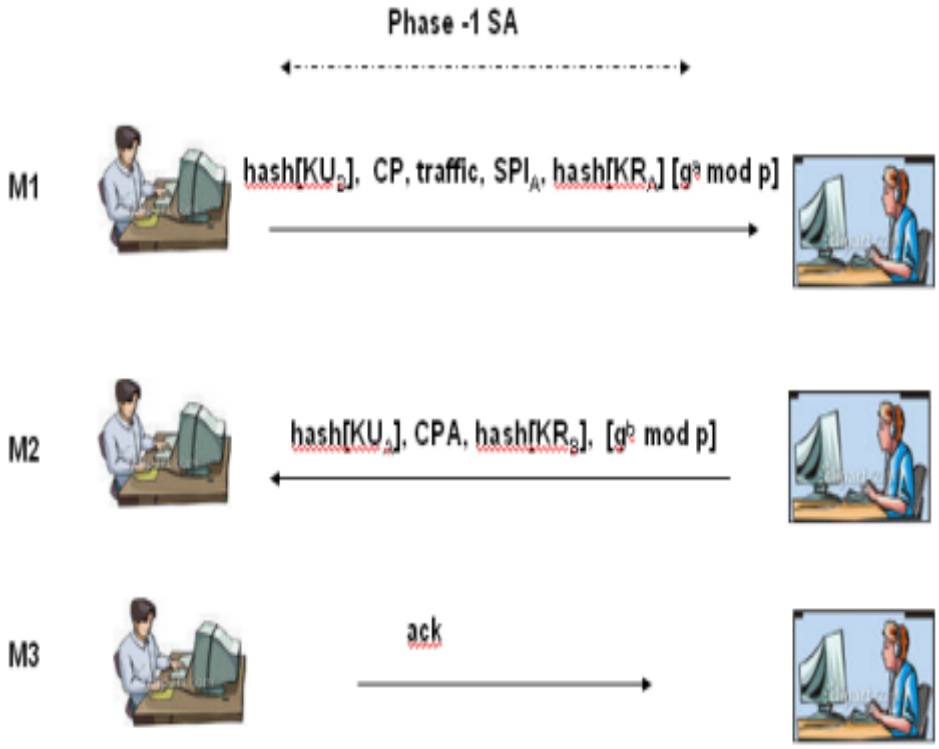

Figure 3.18 La modification de la deuxième phase du protocole IKE en mode rapide (Nagalakshmi et al., 2011)

3.8 Conclusion

Dans ce chapitre, nous avons examiné les successeurs du protocole IKE tel qu'IKEv1, IKEv2, IKE proposé par Haddad et al. en 2004 et IKE proposé par Ray et al. en 2012 afin de les comparer à nos contributions proposées dans le chapitre 5. Dans le chapitre 5, nous présentons nos protocoles proposés pour résoudre certains problèmes de sécurité et d'efficacité du protocole IKE et rendre le protocole IPSec plus sécurisé et plus efficace.

Chapitre IV : Les approches de mesure de sécurité et les métriques

Lord Kelvin a dit « *si vous ne pouvez pas mesurer, vous ne pouvez pas améliorer* ». Donc, pour obtenir des preuves sur le niveau de sécurité d'un système ou des services, une approche de mesure de sécurité est nécessaire. Dans ce chapitre nous présentons les différentes définitions de métriques et mesure de sécurité existantes dans la littérature. Anis que les approches de mesure sécurité existantes. Nous terminons ce chapitre par une contribution basé sur le travail proposé par (Savola and Abie; 2009).

4.1. Les métriques de sécurité

4.1.1. Définition des métriques et des mesures

Pour comprendre les métriques de sécurité, nous devons clarifier la différence entre les métriques et les mesures. La mesure est le processus par lequel des chiffres ou des symboles sont affectés à des attributs d'entités du monde réel, de telle sorte qu'il les décrive en fonction de règles clairement définies. Le but ultime de mesure de la sécurité est de pouvoir obtenir des preuves sur le niveau de sécurité opérationnel. Tandis que les métriques sont obtenues en comparant deux ou plusieurs mesures prises au fil du temps avec un niveau de référence prédéterminé (Savola and Abie, 2009).

En d'autres termes, les mesures sont des données brutes objectives et les métriques sont soit des interprétations humaines objectives ou subjectives de ces données. Le cadre d'évaluation de la sécurité bien développé et les mesures dérivées peuvent agir comme un outil efficace de gestionnaire de sécurité afin de distinguer l'efficacité de sécurité des diverses composantes d'un système, d'un produit ou d'un processus (Shirley, 2006). Comme il a été mentionné précédemment, les métriques de sécurité ont de nombreuses interprétations. Voici quelques unes des élaborations courtes de métriques et mesures de sécurité à long terme.

- Selon (Swanson et al., 2003), les métriques sont des outils conçus pour faciliter la prise de la décision et améliorer la performance grâce à la collecte et l'analyse des données liées à la performance. Le but de la mesure de performance est de surveiller l'état des activités mesurées et de faciliter l'amélioration de ces derniers;

- Selon (Stoddard et al., 2005), les métriques sont les composantes clés de la gestion des risques. Une métrique est une mesure qui est comparée à une échelle ou une référence pour produire un résultat significatif;
- Selon (Xenos, 2006), une métrique est une affectation empirique d'une valeur à une entité visant à décrire une caractéristique spécifique de cette entité;
- Selon (SSE-CMM, 2008), les métriques sont des mesures quantifiables de certains aspects du système ou d'une entreprise. Pour une entité (système, produit, ou autre) où la sécurité est un concept significatif, il y a quelques attributs identifiables qui caractérisent collectivement la sécurité de cette entité. En outre, une métrique de sécurité (ou combinaison de métriques de sécurité) est une mesure quantitative de l'attribut de l'entité. Une métrique de sécurité peut être construite à partir des mesures physiques de niveau inférieur;
- Selon (Hayden, 2010), une métrique est une norme de mesure. Les métriques sont le résultat et la mesure est une activité. La mesure est définie comme l'acte de juger ou d'estimer les qualités de quelque chose, y compris les qualités à la fois physiques et non physiques, par comparaison à autre chose;
- Selon (Hallberg et al., 2011), une métrique de sécurité contient trois parties principales: une amplitude, une échelle et une interprétation. Les valeurs de sécurité des systèmes sont mesurées selon une amplitude spécifiée et à une échelle associée. L'interprétation impose le sens des valeurs de sécurité obtenues;
- Selon (Jansen, 2009), une métrique implique généralement un système de mesure basé sur des mesures quantifiables. Une méthode de mesure utilisée pour déterminer la quantité d'une unité peut comporter un instrument de mesure, la matière de référence, ou le système de mesure.

4.1.2. Les propriétés de métriques de sécurité

Les propriétés de métriques de sécurité peuvent être étudiées en fonction de la classification suivante (Savola and Abie, 2009; Vaughn et al., 2002):

- Les métrique quantitatives vs qualitatives: le résultat des métriques de sécurité peut être soit de nature quantitative ou qualitative. Les métriques quantitatives

sont plus souhaitables que qualitatives, parce qu'il est difficile de trouver des indicateurs quantitatifs qui représentent des phénomènes de sécurité de l'information (Savola and Abie, 2009; Vaughn et al., 2002);

- Les métriques objectivées vs subjectivées: le résultat des métriques de sécurité doit être soit de nature objective ou subjective. Les métriques de sécurité objectivées représentent l'état de sécurité d'un système ou d'un processus à certains niveaux discrets telles que: faible, moyen et élève. Les métriques subjectives prennent généralement en considération les aspects du comportement humain dans la sécurité (Savola and Abie, 2009; Vaughn et al., 2002);
- Les métriques directes vs indirectes: selon la norme standard ISO/IEC 9126, une métrique directe est une mesure d'un attribut atomique du système dans le sens où l'attribut mesuré ne dépend pas des autres attributs. D'autre part, une métrique indirecte est dérivée de mesures d'un ou plusieurs autres attributs (Savola and Abie, 2009; Vaughn et al., 2002);
- Les métriques statiques vs dynamiques: le résultat des métriques dynamiques sera affecté par le temps écoulé alors que les métriques statiques ne prennent pas le temps en compte (Savola and Abie, 2009; Vaughn et al., 2002);
- Les métriques absolues vs relatives: une métrique absolue est de nature atomique dans le sens où elle ne dépend pas de la sortie de toute autre métrique (Savola and Abie, 2009; Vaughn et al., 2002).

4.1.3. L'utilisation des métriques de sécurité

Elles peuvent être utilisées pour aide à la décision, en particulier en matière d'évaluation, de surveillance et de prévision. L'objectif des métriques de sécurité peut comprendre un système technique, un service, un produit, ou une organisation. Nous présentons ci-dessus les façons dont les métriques peuvent être utilisées (Savola and Abie, 2009):

- Les activités de gestion des risques pour les risques de sécurité d'atténuation;
- La comparaison de mécanisme de sécurité;
- Obtenir des informations sur la posture de sécurité d'une organisation, un processus ou un produit;

- ❖ L'assurance de la sécurité (analyse, tests, suivi) d'un produit, d'une organisation ou d'un processus;
- ❖ Test de sécurité fonctionnel d'un système;
- ❖ La certification et l'évaluation d'un produit ou d'une organisation;
- ❖ La surveillance de la sécurité adaptative pendant le fonctionnement du système;
- ❖ La détection et la prévention d'intrusion dans un système.

4.1.4. Les objectifs de mesure de sécurité

En ingénierie de sécurité, l'exactitude de sécurité, l'efficacité et l'efficience de sécurité peuvent être considéré comme les principaux objectifs de mesure. Elles peuvent être définies de la manière suivante (Savola and Abie, 2009):

- ❖ L'exactitude de sécurité représente l'assurance que les mécanismes de la sécurité des applications ont été correctement mises en œuvre dans le système étudié et les interfaces, les données traitées répondent aux exigences de sécurité. L'exactitude de sécurité peut être considérée comme un objectif de qualité de sécurité et une condition nécessaire mais pas suffisante pour les deux objectifs (haut niveau) de la mesure, telle que l'efficacité et l'efficience de la sécurité (Savola and Abie, 2009);
- ❖ L'efficacité de la sécurité représente l'assurance que les mécanismes de sécurité du système répondent aux objectifs de sécurité énoncée et que les attentes de la résilience dans l'environnement d'utilisation sont satisfaites (Savola and Abie, 2009);
- ❖ L'efficience de la sécurité représente l'assurance que la qualité de la sécurité adéquate a été réalisée dans le système étudié (Savola and Abie, 2009);
- ❖ Les exigences de sécurité sont les exigences de qualité qui spécifie:
  - ➢ Une quantité de sécurité requis en termes de critère spécifique au système;
  - ➢ Un niveau minimum d'une mesure de qualité associée qui est nécessaire pour répondre à une ou plusieurs politiques de sécurité.

4.2. Les dimensions mesurables

L'Union internationale des télécommunications (UIT) a défini un ensemble plus vaste de la dimension de sécurité telle que: le contrôle d'accès, l'authentification, la non-répudiation, la confidentialité des données, la sécurité des communications, l'intégrité des données, la disponibilité et la vie privée. Plusieurs autres facteurs influent sur la sécurité des systèmes d'information, comme la responsabilisation, la vérification, la contrôlabilité, l'exactitude, l'identification, la récupération, la fiabilité, la robustesse, la sécurité et la supervision, ainsi que la fonctionnalité. La sécurité, la confiance, la fiabilité et la confidentialité sont souvent regroupées lors de la définition des objectifs relevant de la sécurité des systèmes et des services (ITU, 2003).

4.3. Les travaux connexes

Nous allons discuter les approches connexes d'évaluation de la sécurité du réseau. Wang et Wulf (Wang et Wulf, 1996) ont proposé une approche de mesure de sécurité d'un système de sécurité. Ils ont décomposé un système de sécurité en des composants mesurables selon la dimension de sécurité. La référence (Wang et Wulf, 1996) présente la décomposition de l'authentification, la confidentialité et l'intégrité. La Figure 4.1 illustre la décomposition hiérarchique de l'authentification. Les feuilles de la décomposition hiérarchique sont les éléments mesurables de l'authentification.

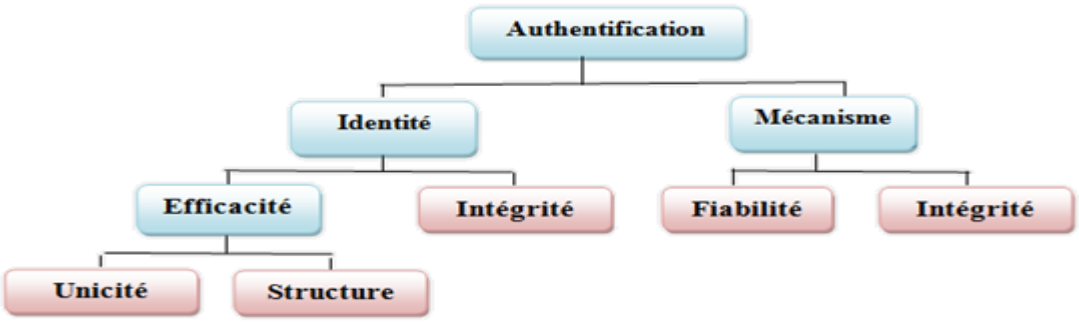

Figure 4.1 La décomposition de l'authentification (Wang et Wulf, 1996)

(Savola et Abie, 2009) ont proposé une taxonomie des mesures de sécurité basée sur l'étude de (Wang et Wulf, 1996), où ils ont donné des détaillées concernant l'évaluation de chaque élément mesurable de dimension de sécurité. Pour chaque dimension, ils ont donné les exigences de la dimension; puis, ils ont définit les éléments mesurable de chaque dimension. Enfin, ils ont proposé une formulation ou une méthode de mesure pour chaque élément mesurable. Dans la section 4.4 nous implémentant les métriques proposées par (Savola et Abie, 2009) sur plusieurs systèmes d'authentification.

Ahmed et al. (Ahmed et al., 2008) ont proposé une approche de mesure de sécurité d'un réseau. Ils ont identifié et quantifié objectivement les facteurs de risque en matière de sécurité, ce qui englobe les vulnérabilités existantes, la tendance historique des vulnérabilités des services accessibles à distance, la prévision des vulnérabilités potentielles pour tout service de réseau général et leur gravité estimée, et aussi la propagation d'une attaque dans le réseau.

Dans (leon and Saxena, 2010), les auteurs ont proposé un nouveau cadre pour mesurer et évaluer les différents aspects de la sécurité. Le cadre repose sur un grand nombre des caractéristiques de risque actuel, la gestion des IT et les normes de sécurité.

Casola et al. (Casola et al., 2007) ont proposé une méthodologie pour évaluer le niveau d'un système de sécurité par la définition des métriques de sécurité qui se base sur la représentation et la comparaison entre les différentes politiques de sécurité.

Atzeni et al. (Atzeni et al., 2005) ont discuté l'importance de l'évaluation de la sécurité des systèmes informatiques, et ils ont présenté une liste des métriques de sécurité.

Récemment, Krautsevich et al. (Krautsevich et al., 2010) ont présenté une description formelle des métriques de sécurité. Ils ont formalisé un certain nombre de métriques de sécurité telles que le nombre d'attaques, le coût minimal d'attaque, le coût minimal pour la réduction des attaques, la plus courte longueur d'attaque, la probabilité maximale d'attaque, la surface d'attaque, etc.

Une nouvelle approche pour mesurer la sécurité du système qui peut calculer le risque résiduel de chaque vulnérabilité est décrite dans (Sahinoglu, 2005). Bartol et al. (Bartol et al., 2009) ont discuté l'importance de la quantification d'une organisation pour mesurer les dimensions de sécurité.

4.4. Notre apport pour l'implémentation des métriques d'authentification

Dans cette section nous présentons les métriques d'authentification proposés par (Savola and Abie, 2009). Et nous implémentons ces métriques sur plusieurs systèmes d'authentification tel que le certificat numérique, système par mot de passe et système acoustique-anatomique. Cette implémentation nous permettra de quantifier la sécurité et nous pouvons par la suite effectuer des comparaisons de différentes métriques d'une part et des différents systèmes d'autre part. Les travaux de Savola et Abie (Savola and Abie, 2009) constituent pour nous un point de départ dans un domaine de recherche à savoir les métriques de sécurité pour le moment très jeune et très prometteur.

- **L'unicité de l'identité**: c'est le nombre d'informations utilisées pour identifier l'identité divisée par le nombre total d'informations, plus un facteur de pondération multiplié par le nombre d'informations où la condition d'unicité n'est pas vérifiée (Savola and Abie, 2009).
  - Facteur de pondération: c'est le nombre d'informations où la condition d'unicité n'est pas vérifiée sur le nombre d'informations total utilisé pour identifier l'identité.

$$w_{NSID} = \frac{NSID}{ID} \quad (1)$$

- **La structure de l'identité (SIA)**: la structure de l'identité dépend de la structure des paramètres et la qualité de sécurité.

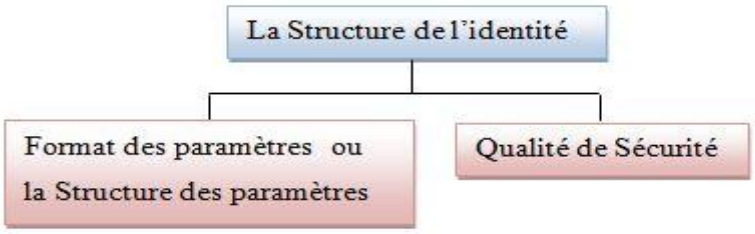

Figure 4.2 Décomposition de la structure d'identité basée sur la proposition de (Savola and Abie, 2009)

- Format des paramètres ou la structure des paramètres (SP): dépend de l'atomicité des paramètres. Ces paramètres peuvent être atomiques ou composés.

  Dans le cas composé, les paramètres sont mal structurés, par contre dans le cas atomique, ils sont bien structurés. Par conséquent, le système d'authentification nécessite une méthode de décomposition pour vérifier ces paramètres; ce qui produit une perte de temps et peut causer une augmentation du taux d'erreur.

  - L'atomicité des paramètres (AP): c'est le nombre de paramètres atomiques divisé par le nombre total des paramètres utilisés.

$$AP = \frac{n(AMT)}{n(ID)} \quad (2)$$

- Pour l'évaluation de la qualité de sécurité, on utilise un questionnaire de condition concernant :
  - ✓ La solution de la structure d'identité.
  - ✓ Les algorithmes de cryptage utilisé dans le mécanisme d'authentification.
  - ✓ L'authentification multiple.

Il ya plusieurs façons de transformer un questionnaire en métrique. Dans notre travail on calcule l'effectif de réponse « oui ».

$$QS = \frac{n(RO)}{n(ID)} \quad (3)$$

Donc :

$$SIA = \frac{1}{2} [QS + SP(\text{ou } AP)] \quad (4)$$

❖ **L'intégrité de l'identité**: l'intégrité de l'identité a toujours un rôle central dans les systèmes d'authentification. L'altération non autorisée de l'identité, si elle n'est pas évitée, elle peut causer des résultats catastrophiques tels que les faux rejets et les fausses acceptations (Wang et Wulf, 1996).

$$IIA = \frac{1}{2} (PCI+SPI) \quad (5)$$

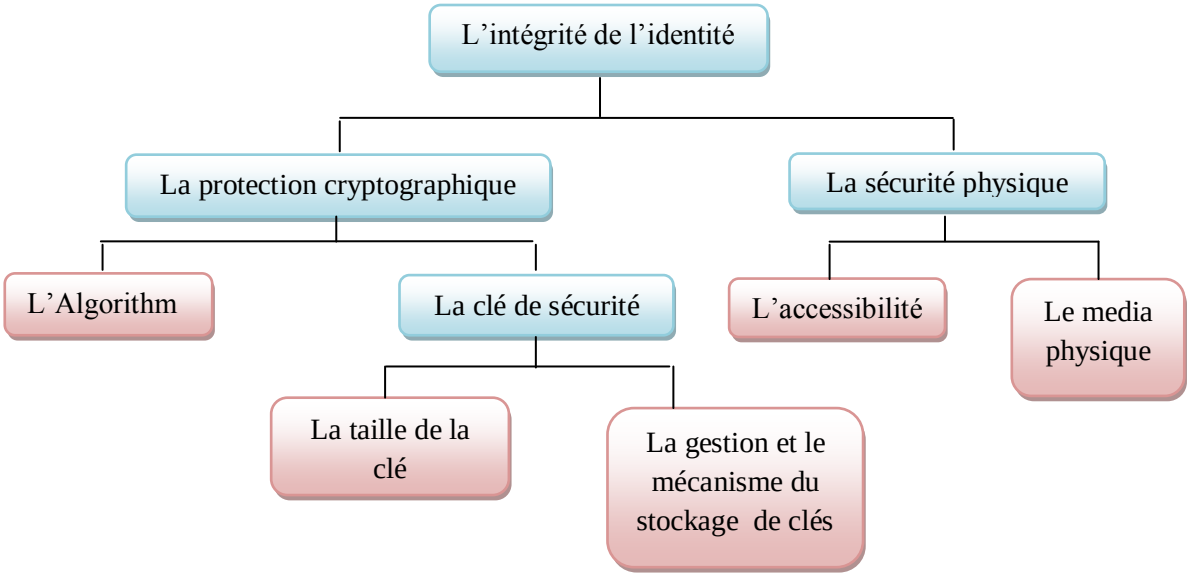

Figure 4.3 Notre décomposition de l'intégrité de l'identité basée sur le travail de (Wang et Wulf, 1996)

➢ **La protection cryptographique (PCI)**: consiste à sécuriser les données. Elle dépend de l'algorithme utilisé pour le chiffrement et le déchiffrement des données et la taille de la clé, ainsi que le mécanisme de gestion et de stockage de la clé (Wang et Wulf, 1996).

Pour évaluer la protection cryptographique un questionnaire de condition peut solliciter des informations sur la façon dont l'algorithme est rigoureusement testé, et combien de temps a-t-il utilisé, et quel type de cryptanalyse a été effectuée contre lui ? (Wang et Wulf, 1996).

- Pour évaluer la gestion de mécanisme et le stockage de clé un ensemble de questions a été posé concernant:
  - ✓ La manière utilisée pour la gestion de la distribution des clés;

- ✓ Les algorithmes utilisés pour le cryptage des clés;
- ✓ La sécurisation de fichier où on sauvegarde les clés ainsi que les données.

La méthode de mesure utilisée pour l'évaluation comprend :

- **La gestion et le stockage de la clé,** on calcule le pourcentage de nombre de réponses « oui » sur le nombre total des réponses.
- **La vérification de la suffisante de taille de la clé**, on calcule le ratio pour la vérification.

$$\text{Ratio} = \frac{\textit{taille de la clé utlisée}}{\text{taille maximum de clé qui peut être utilisée}} \quad (6)$$

Donc : $$PCI=\frac{1}{2}(ACP+CSCP) \quad (7)$$

- **La sécurité physique (SPI)**: la sécurité physique traite des aspects de la sécurité de l'ordinateur relatif à la situation physique de la machine, ou son environnement opérationnel.

  La sécurité physique est importante parce que les attaques physiques représentent la moitié des attaques qu'un système informatique subi.

$$SPI=\frac{1}{2}(MPSP+ASP) \quad (8)$$

- Dans notre travail nous supposons que le media physique respecte toutes les bonnes propriétés.
- **L'accessibilité (ASP)**: l'accès aux ressources informatiques peut devenir impossible pour plusieurs raisons telles que:
  - Les catastrophes naturelles;
  - Les attaques telles que le déni service.

Ce qui engendre un arrêt du fonctionnement du centre de traitement. Ceci nous amène à faire une petite étude préventive contre ces risques:

Les actions préventives:

- ✓ On installe nos systèmes là où les risques d'une catastrophe naturelle est réduit.
- ✓ On met en œuvre un mécanisme de protections contre les intrusions.
- ✓ On met en place une solution de prise de contrôle à distance.
  - Pour évaluer l'accessibilité de sécurité physique on doit calculer le pourcentage des réponses « oui » aux questions concernant la prévention.

❖ **L'intégrité du mécanisme**: l'intégrité des composantes d'un mécanisme d'authentification est une condition préalable à l'intégrité des données, qui à son tour indique que les données n'ont pas été modifiées ou détruites de façon non autorisée. Le système et ses composants doivent générer, traiter, conserver ou transmettre les données de telle sorte que l'intégrité des données est préservée. Les erreurs d'intégrité sont au cœur des mesures d'intégrité et comprennent généralement l'altération, la suppression, l'adjonction, la perte des données (Savola and Abie, 2009).

$$\text{IMA}= \frac{\text{n(AU)}}{n(AU)+w_{IEam} \quad \text{n}(IE_{AM})} \quad \text{(Savola and Abie, 2009)} \qquad (9)$$

❖ **La fiabilité du mécanisme d'authentification**: la fiabilité du mécanisme d'authentification dépend du temps passé sur l'ingénierie des exigences de l'authentification, le temps passé pour les tests du mécanisme d'authentification, et les fonctionnalités d'authentification adaptative. Dans notre étude on a fixé la fiabilité.

4.5. Les systèmes étudiés

Les systèmes d'authentification que nous avons mesurés sont:

❖ **L'identification acoustique-Anatomique du locuteur**

L'architecture de système d'authentification de paramétrisation acoustique-Anatomique

1. L'information acoustique est extraite et modélisée sous forme de distribution de possibilistes;

2. Les distances anatomiques exploitées sont à leur tour représentés par des distributions de possibilistes;
3. La fusion possibiliste nous permet de fusionner des possibilistes obtenues dans les étapes précédentes ce qui produit une distribution fusionnée;
4. La « défuzzification » nous permet d'obtenir le vecteur acoustique anatomique représentatif du locuteur;

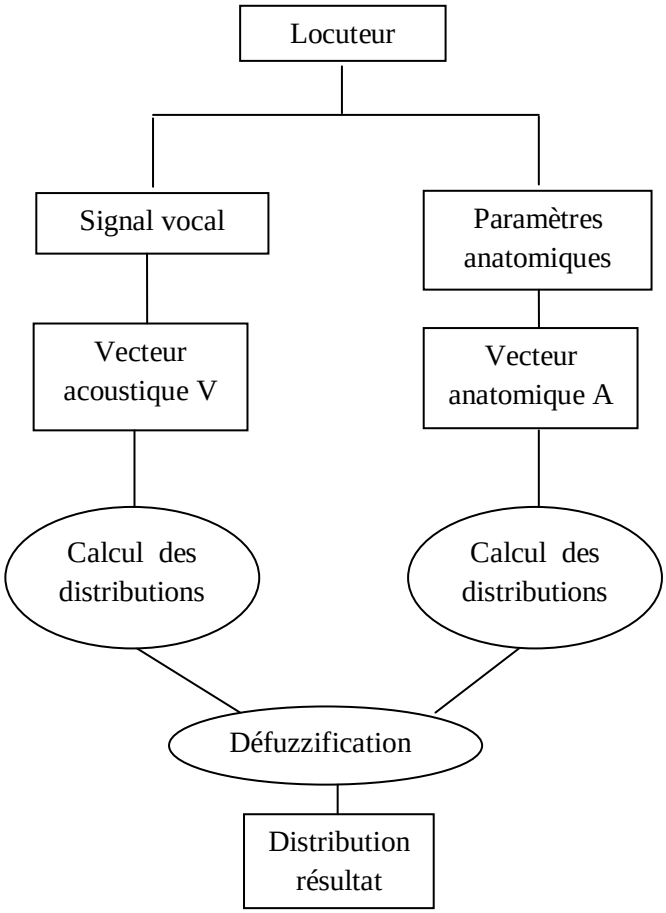

Figure 4.4 La paramétrisation du locuteur (Debbeche and Ghoualmi, 2008)

- Le vecteur de paramétrisation représente les données pour authentifier le locuteur;
- Aucun algorithme de cryptage n'est utilisé dans ce système;
- La base de données n'est pas sécurisée;
- À l'instant t=0 le système n'est pas en panne.

❖ **Le système qui utilise « mot de passe-nom utilisateur »**

Le nom d'utilisateur et le mot de passe sont deux méthodes dans la catégorie des informations connus, où la solution d'identité est numérique, les formats des données sont atomiques. Nous supposons dans ce système proposé que:

- Le nom d'utilisateur et le mot de passe sont stockés d'une manière sécurisée et cryptée à l'aide « RSA » avec une clé de 2048 bits;
- L'accès à la base de données est avec mot de passe;
- L'algorithme de cryptage est bien implémenté;
- On installe le système là où les risques d'une catastrophe naturelle sont réduits;
- Mettre en œuvre des protections contre l'intrusion;
- Le support physique : respecte les bonnes propriétés;
- L'utilisateur doit saisir son mot de passe et son nom d'utilisateur à partir d'un clavier numérique;
- La gestion de la distribution de la clé : un certificat public;
- L'algorithme pour vérifier l'intégrité d'identité est bien implémenté ;
- À l'instant t=0, le système n'est pas en panne.

❖ **Le système qui utilise le «certificat numérique »**

Les attribues du certificat numérique sont:

- La version X.509 à laquelle le certificat correspond;
- Le numéro de série du certificat;
- L'algorithme de chiffrement utilisé pour signer le certificat;
- Le nom de l'autorité émettrice de certification;
- La date de fin de validité du certificat;
- L'objet de l'utilisation de la clé publique;
- La clé publique du propriétaire du certificat;
- La signature de l'émetteur du certificat.

EXEMPLE

Informations
Autorité de certificat: version Nom du propriétaire: xxx Email : xxxxxxxxx Validité : 04/10/2011 au 04/10/2011 Clé publique:10.5a.55bbb Algorithme: RSA Numéro de série
Signature

Nous supposons dans le système proposé que:

- L'algorithme utilisé pour le cryptage est RSA;
- Les informations de certificats sont toutes atomiques;
- L'algorithme de cryptage est bien implémenté;
- On installe le système, là où les risques d'une catastrophe naturelle sont réduits;
- La mise en œuvre des protections contre l'intrusion;
- Le support physique respecte les bonnes propriétés;
- La gestion de la distribution de la clé : un « certificat public »;
- L'algorithme pour vérifier l'intégrité d'identité est bien implémenté;
- À l'instant t=0, le système n'est pas en panne.

❖ **Les autres systèmes sont des combinaisons des systèmes précédents**

- Un système d'identification automatique du locuteur et certificat numérique;
- Un système d'identification automatique du locuteur et mot de passe-nom d'utilisateur;
- Un système de mot de passe-nom utilisateur et certificat numérique.

4.5.1. L'implémentation des métriques d'authentification

Nous implémentons les métriques que nous avons présentées et définies précédemment sur les différents systèmes cités ci-dessus.

4.5.1.1. Les environnements de programmation

Nous présentons, dans ce qui suit, l'environnement logiciel et matériel utilisé dans le cadre de ce travail.

4.5.1.1.1. L'environnement logiciel

Le langage de programmation et l'IDE (environnement de développement intégré) utilisé pour le développement de notre application sont:

- Java Sun jdk 1.6;
- Éclipse europa.

Le modèle de programmation orientée objet a été choisi afin d'assurer une meilleure modularité et portabilité du code. Le fait que l'algorithme est développé en un langage interprété permet sa réutilisabilité par tous les systèmes d'exploitation sans régénérer aucun code.

4.5.1.1.2. L'environnement matériels

La plateforme de programmation et de test est un :

PC portatif doté de :

- Processeur Intel(R) Core(TM) 2 Duo CPU T5870 © 2.00GHz 2.00 GHz;
- Mémoire RAM 2,00 Go DD2;
- Carte graphique 1,00 Go;
- Disque Dur de 320 Go;
- Écran 17 pose;
- Système d'exploitation « Windows 7 » Edition Familiale Premium Service Pack 1.

4.5.1.1.3. Les classes utilisées

Parmi les différentes classes codées, on trouve les classes clés de notre application qui sont:

- **CalculerBMCs**: cette classe permet de calculer les différentes métriques d'authentification telles que l'unicité de l'identité, la structure de l'identité,

l'intégrité de l'identité, et l'intégrité de mécanisme que nous avons présenté précédemment;

- **NewJFrame**: cette classe représente la classe principale qui s'occupe de l'affichage de notre interface.

4.5.1.1.4. Les interfaces du système développé

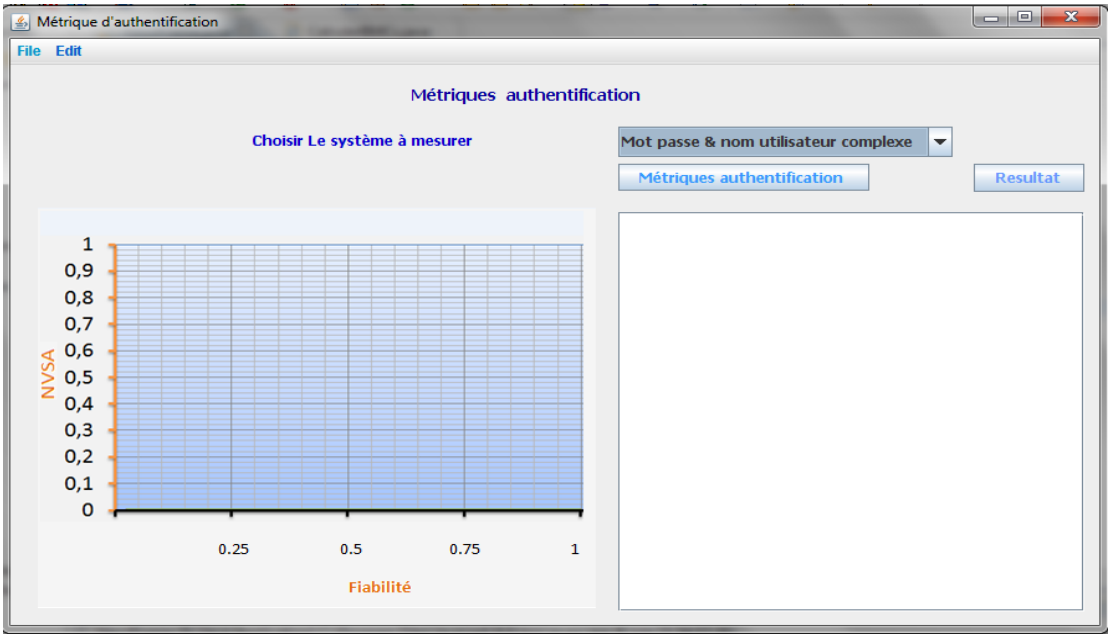

Figure 4.5 L'interface de l'implémentation

L'utilisateur qui veut mesurer le niveau global d'authentification doit tout d'abord sélectionner, qu'est ce qu'il veut mesurer exactement, parmi les systèmes qui ont été proposés. Après, il doit cliquer sur le bouton « ***Résultat*** », où les résultats seront affichés dans un composant « Jliste», ainsi qu'une courbe. Cette courbe représente le niveau de sécurité global d'authentification par rapport à la fiabilité de mécanisme.

Si on clique sur le bouton « ***Métrique d'authentification*** » nous aurons la fenêtre suivante, qui contient la décomposition d'authentification, et les métriques utilisées pour mesurer le niveau de sécurité globale de la solution d'authentification.

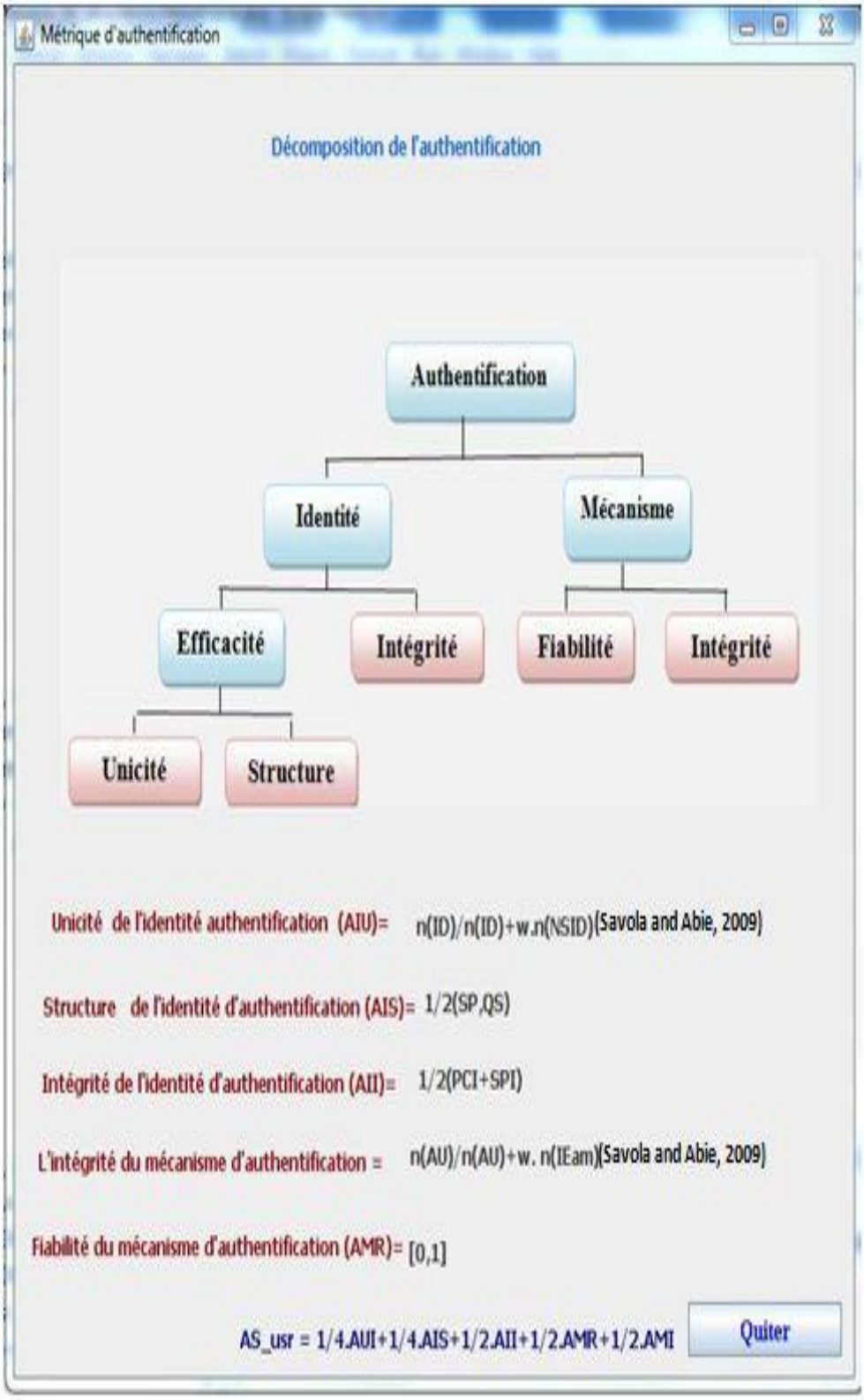

Figure 4.6 Les métriques d'authentification implémentées

❖ **Les résultats du système « Identification automatique du locuteur »**

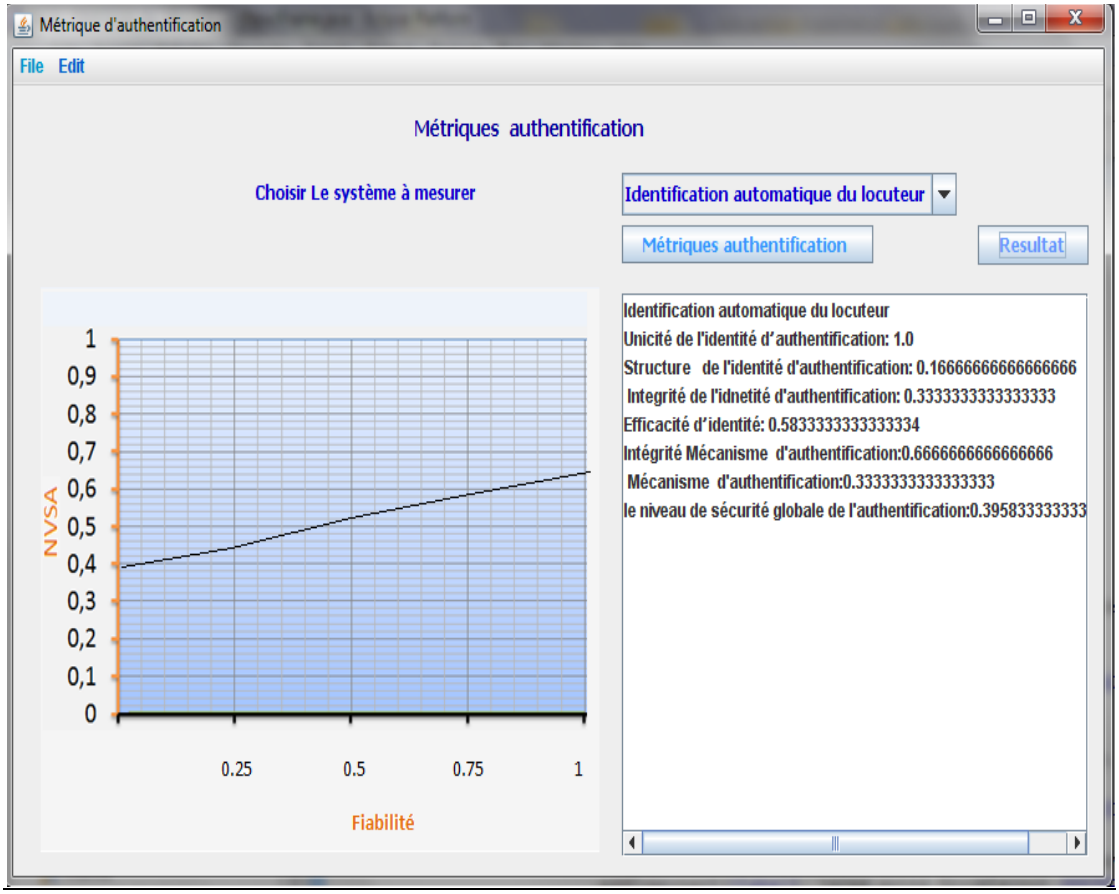

Figure 4.7 Les résultats du système identification automatique du locuteur

$$AS=\frac{1}{4}UIA+\frac{1}{4}SIA+\frac{1}{2}IIA+\frac{1}{2}IMA+\frac{1}{2}FMA$$

- L'unicité de ce mécanisme est de 1, car chaque utilisateur est représenté par un vecteur de paramètre unique;
- La structure identité = 0.16 parce qu'on a aucun algorithme de cryptage de données, et les paramètres utilisés sont composés;
- L'intégrité de l'identité est faible parce qu'aucun algorithme de chiffrement n'est utilisé. De plus, les données sont sauvegardées dans une base de données non sécurisée;
- L'intégrité du mécanisme: nous avons supposé que nombre d'erreurs d'intégrité égale à 10 erreurs, et le facteur de pondération est fixé à 0.5; donc, on trouve l'intégrité de mécanisme égale à 0.6;

Le niveau global d'authentification dépend de la fiabilité du mécanisme d'authentification avec un pourcentage de 25%.

❖ **Les résultats du système « mot de passe et nom utilisateur »**

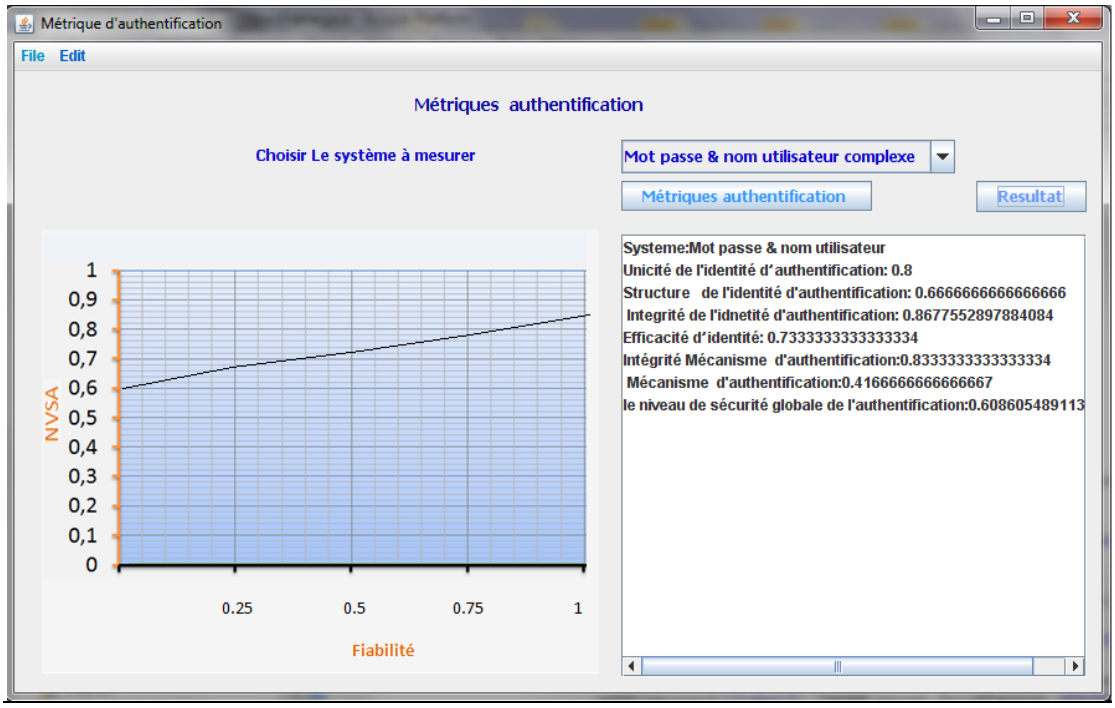

Figure 4.8 Les résultats du système « mot de passe et nom utilisateur »

Dans ce système:

- L'unicité est de 0.8 car on peut avoir au minimum deux personnes qui possède le même nom d'utilisateur;
- La structure de l'identité est de 0.66 parce que les paramètres sont bien structurés et plusieurs techniques de sécurité contre plusieurs attaques ont été utilisées;
- L'efficacité de l'intégrité est élevée parce que ce système utilise des algorithmes de chiffrement qui permettent de protéger les informations confidentielles contre plusieurs types d'attaques. De plus les données utilisées sont sauvegardées dans une base de données protégée;
- Par l'application du formule proposé par Savola et Abie (Savola and Abie, 2009) l'intégrité du mécanisme est de 0.83.

❖ **La combinaison entre les systèmes (mot de passe, identification automatique du locuteur, et certificat numérique)**

➢ Le système d'identification automatique du locuteur et certificat numériques

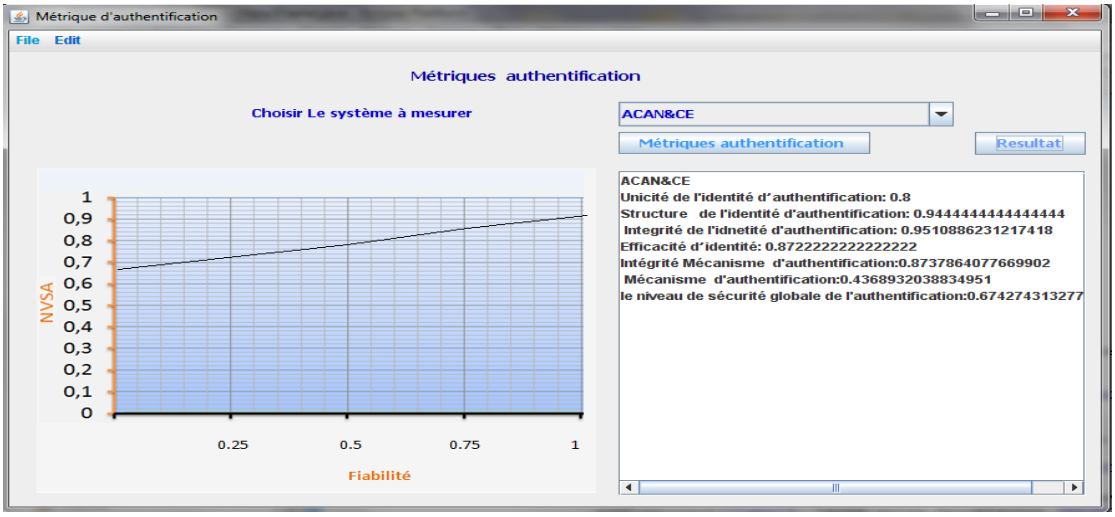

Figure 4.9 Les résultats du système d'identification automatique du locuteur et certificat numérique

On suppose que le nombre des erreurs d'intégrité pour le système l'identification automatique du locuteur et certificat numérique est égal à 5.

Le niveau de sécurité a augmenté parce que l'unicité et la structure d'identité sont élevées ainsi que l'intégrité du mécanisme.

➢ Le système de mot de passe-nom utilisateur et certificat numérique:

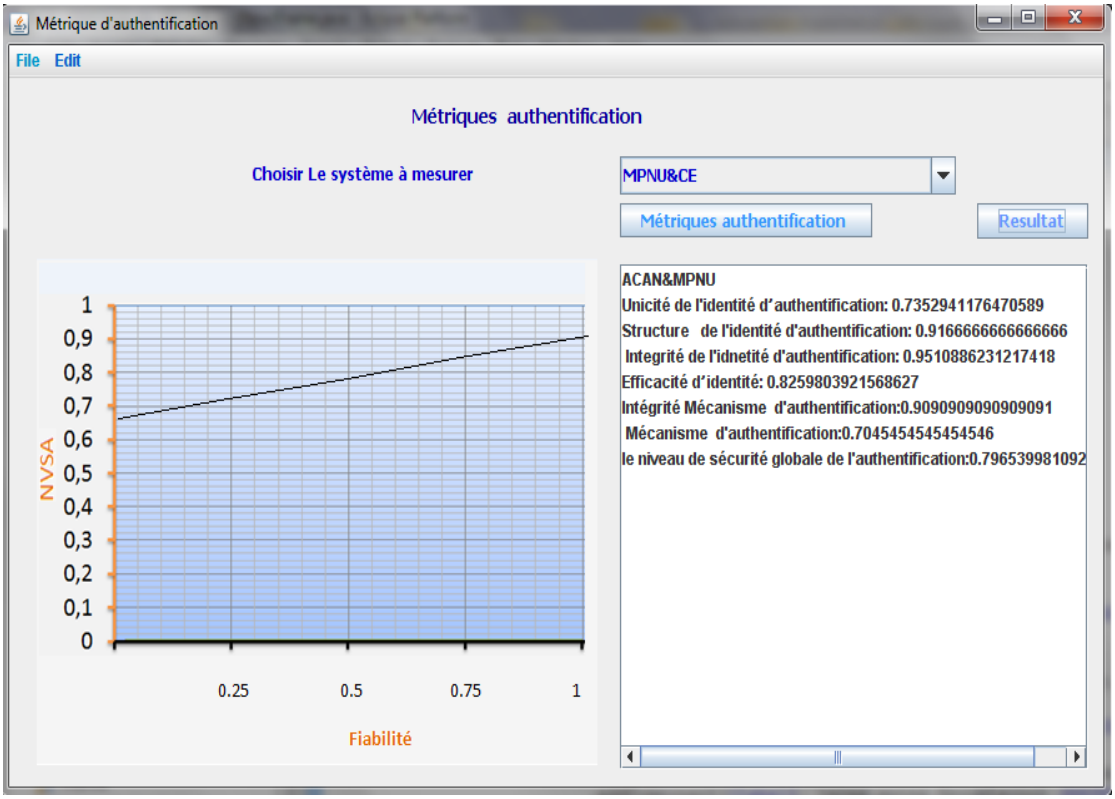

Figure 4.10 Le système de mot de passe-nom d'utilisateur et certificat numérique

La combinaison entre les mécanismes et une solution physique bien structurée est le meilleur système d'authentification qui donne de bons résultats pour le système d'authentification. L'unicité de l'identité et la structure des paramètres utilisés ont une influence sur la classification. On peut déduire que le choix du système d'authentification sécurisé dépend de l'unicité de l'identité et de la structure des paramètres et bien sûr l'intégrité de l'identité.

4.6. Les apports de notre approche par apport au travail de (Savola and Abie, 2009)

Métriques	(Savola and Abie, 2009)	Notre travail
L'unicité de l'identité	UIA=$\frac{\mathrm{n(ID)}}{n(ID)+w_{NSID}\quad \mathrm{n(NSID)}}$	UIA= $\frac{\mathrm{n(ID)}}{n(ID)+w_{NSID}\quad \mathrm{n(NSID)}}$; w_{NSID} =$\frac{NSID}{ID}$.
La structure de l'identité	peut être «élevée», «moyen» ou «faible».	SIA = $\frac{1}{2}$ (SP+QS); SP= $\frac{n(AMT)}{n(ID)}$; QS = $\frac{n(RO)}{n(ID)}$.
L'intégrité de l'identité	IIA = f (AIS, IEID) où f est une fonction non spécifié.	IIA= $\frac{1}{2}$ (PCI+SPI); PCI=$\frac{1}{2}$ (ACP+CSCP); SPI=$\frac{1}{2}$ (MPSP+ASP).
L'intégrité de mécanisme	AMI= $\frac{\mathrm{n(AU)}}{n(AU)+w_{IEam}\quad \mathrm{n}(IE_{AM})}$	IMA= $\frac{\mathrm{n(AU)}}{n(AU)+w_{IEam}\quad \mathrm{n}(IE_{AM})}$
La fiabilité de mécanisme	AMR = f (T_{AMRreq} , $T_{AMRtest}$, T_{RMRasm}) où f est une fonction non spécifié.	[0,1]

Tableau 4.1 Les apports de notre approche par rapport au travail de (Savola and Abie, 2009)

4.7. Conclusion

Dans ce chapitre nous avons présenté les différents concepts du métrique de sécurité tels que: les différentes définitions de métrique et de mesure existantes dans la littérature, les propriétés des métriques de sécurité, les objectifs de la mesure de sécurité et les dimensions mesurables. Nous avons étudié les différentes approches

de mesure de sécurité proposée dans la littérature afin de proposer une approche qui permet de mesure la sécurité de l'IPSec.

En outre, nous avons proposé une contribution en nous basant sur le travail de (Wang and Wulf, 1996) et (Savola and Abie, 2009). Cette contribution a été présentée dans la conférence l'Optimisation et les Systèmes d'Information, Béjaia, Algérie (2014).

Chapitre V: Nos contributions

Dans les chapitres précédents, nous avons étudié le protocole d'IPSec et les métriques de sécurité afin d'améliorer la sécurité de ce protocole. Dans notre thèse, nous avons concentré notre effort sur la phase de l'initialisation du protocole IPSec qui représente le cœur de l'architecturer d'IPSec où nous avons examiné la sécurité des différents protocoles existants dans la littérature en utilisant la métrique « le nombre d'attaques ». Dans ce chapitre, nous présentons nos trois contributions pour le protocole IKE qui permettent d'améliorer la sécurité du protocole IPSec; puis, nous proposons une nouvelle approche d'évaluation de sécurité basée sur les politiques de sécurité.

5.1. La première proposition pour améliorer le protocole IKE

Dans cette section, nous introduisons la première proposition du protocole IKE où nous avons focalisé nos recherche sur la résistance contre les attaques suivantes: DoS, l'homme du milieu et par rejeu (Ahmim et al., 2013).

Dans cette proposition, nous avons utilisé le protocole D-H proposé par (Li, 2010) pour développer un nouveau protocole IKE dans le but de renforcer la sécurité du protocole IKE contre attaques DoS, l'homme du milieu (man-in-the-middle) et par rejeu, ainsi que la minimisation du nombre d'échange dans la première et la deuxième phase du protocole IKEv1.

Notre protocole est composé de six messages. Les quatre premiers sont utilisés pour établir SA-IKE et les deux derniers, qui sont sous protection de la clé de session partagée, sont utilisés pour établir SA-IPSec. Contrairement aux travaux connexes, notre protocole peut résister aux différents types d'attaques telles que (DoS, l'homme du milieu, par rejeu), de plus plusieurs propriétés de sécurité sont vérifiées.

5.1.1. Notations utilisées

- ID_a: l'identité de l'initiateur A;
- ID_b: l'identité du répondeur B;
- SA_i: une liste de propositions cryptographiques de l'initiateur (propositions de l'association de sécurité du protocole IKE);
- SA_r: les protocoles cryptographiques sélectionnés par le répondeur de la liste envoyée par l'initiateur (l'association de sécurité choisie du protocole IKE);

- P_A: mot de passe de l'initiateur;
- P_B: mot de passe du répondeur;
- $\oplus$: XOR;
- ||: concaténation;
- N1: nombre aléatoire;
- SA_{ipsec1}: une liste de propositions cryptographiques de l'initiateur (l'association de sécurité proposée d'IPSec);
- SA_{ipsec2} : les protocoles cryptographiques sélectionnés par le répondeur de la liste envoyée par l'initiateur (l'association de sécurité choisie d'IPsec);
- H: fonction de hachage (.);
- K_{AB}: la clé de session obtenue par l'initiateur et le répondeur;
- E_{KAB} (.): cryptage en utilisant un système cryptographique symétrique avec la clé K_{AB} ;
- AS: serveur d'authentification.

5.1.2. La description du protocole

Le protocole IKE proposé entre l'initiateur (Alice) et le répondeur (Bob) en utilisant le protocole D-H (Li, 2010) est représenté sur la figure 5.1. Il se compose de 6 étapes:

- Étape 1:Initiateur⟶Répondeur : SA_i

  L'initiation envoie au répondeur une série de propositions des algorithmes cryptographiques SA-IKE.

- Étape 2: Répondeur ⟶Initiateur : SA_r

  Le répondeur choisit le SA_r de SA_i en fonction de sa préférence et il envoie SA_r à l'initiateur. Si le répondeur n'accepte aucun algorithme existant dans SA_i, il peut rejeter la liste complète de SA et il renvoie une erreur dans le deuxième message à l'initiateur.

- Étape 3 : Initiateur ⟶ Répondeur: Y_A|| $H(Y_a||N1)$ ||$(SA_r||N1)$

  A la réception du message du répondeur, l'initiateur effectue les opérations suivantes:

  - Sélectionne un nombre aléatoire X_a;

- Calcule $Y_a=\alpha^{Xa} \bmod q$;
- L'initiateur (Alice) envoie un message de demande à l'AS qui comprend l'identité de l'initiateur et du répondeur (ID_a, ID_b);
- L'AS: après avoir reçu un message de l'initiateur (ID_a || ID_b), le serveur d'authentification (AS) envoie N1⊕PA à l'initiateur et N1⊕PB au répondeur;
- L'initiateur calcule N1 = N1⊕PA⊕PA et il envoie Y_A|| $H(Y_a||N1)$ ||(SA_r||N1) au répondeur.

❖ Étape 4: Répondeur ⟶ Initiateur: Y_b|| $H(Y_b||N1)$ ||(SA_i||N1)

Lors de la réception d'un message de l'initiateur, le répondeur effectue les opérations suivantes:

- Calcule: N1 = N1⊕PB ⊕PB ;
- Calcule: H'(Y_a|| N1) et H'(SA_r || N1) où Y_a est envoyé par de l'initiateur, N1 de l'AS et SA_r (la sécurité d'association choisie par le répondeur);
- Vérifie si $H(Y_a||$ N1) = H'(Y_a|| N1) et H (SA_r || N1) = H'(SA_r || N1). Si la vérification échoue, le répondeur termine l'exécution; sinon, le répondeur sélectionne un nombre aléatoire X_b, il calcule $Y_b=\alpha^{Xb} \bmod q$ puis il envoie Y_b|| $H(Y_b||N1)$ ||(SA_i||N1) à l'initiateur.

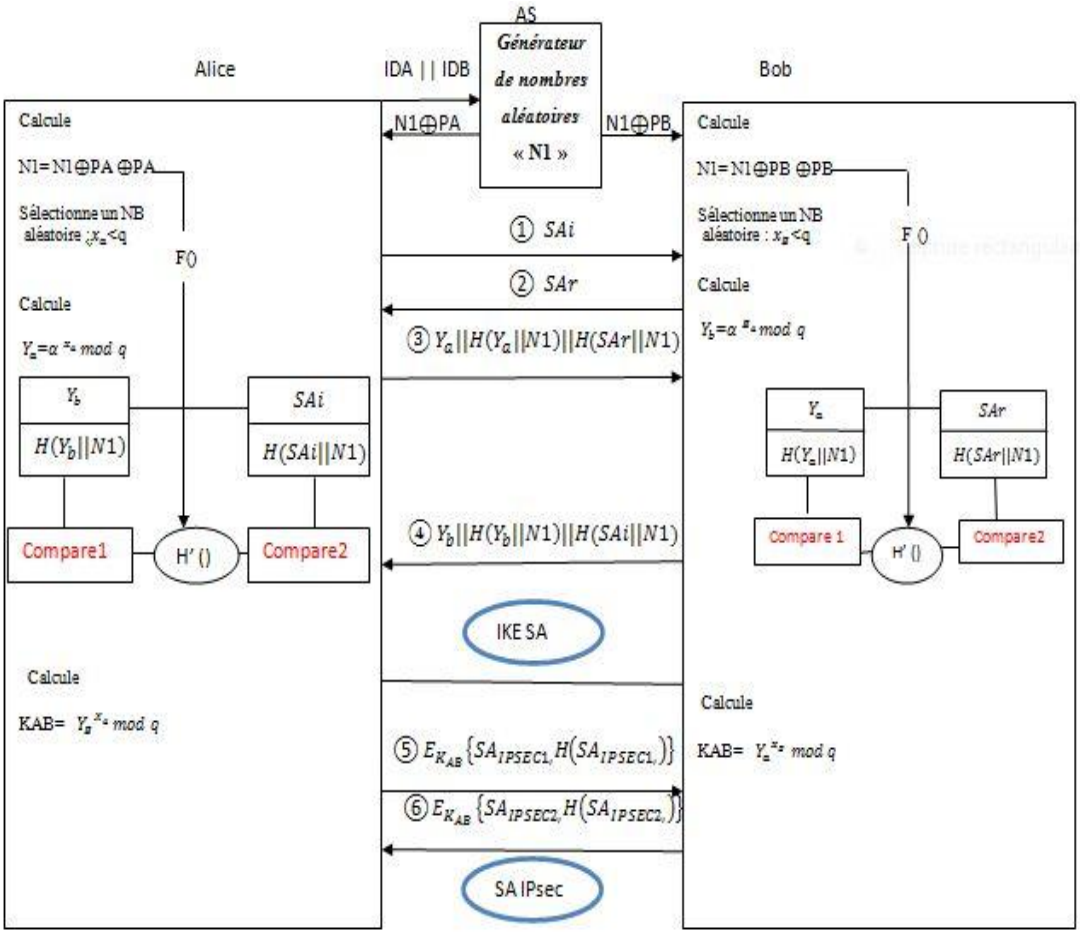

Figure 5.1 Notre protocole IKE (Ahmim et al., 2013)

❖ Étape 5: Initiateur⟶ Répondeur : $E_{KAB}\{SA_{ipsec1}, H(SA_{ipsec1})\}$

L'initiateur effectue les opérations suivantes:

- Calcule: $H'(Y_b||\ N1)$ et $H'(SA_i\ ||\ N1)$ où Y_b est envoyé par le répondeur, N1 de l'AS et SA_i (la sécurité d'association proposée par le l'initiateur).
- Vérifie si $H\ (Y_b||\ N1) = H'\ (Y_b||\ N1)$ et $H\ (SA_i\ ||\ N1) = H'(SA_i\ ||\ N1)$. Si la vérification échoue, l'initiateur termine l'exécution; sinon, l'initiateur calcule K_{AB} et il crypte SA_{ipsec1} et $H(SA_{ipsec1})$ par la clé de cryptage K_{AB} générée précédemment et il l'envoie au répondeur.

❖ Étape 6: Répondeur ⟶Initiateur: $E_{KAB}\{SA_{ipsec2}, H(SA_{ipsec2})\}$

Lors de la réception d'un message de l'initiateur, le répondeur exécute les opérations suivantes:

- Décrypte le message crypté reçu par K_{AB};

- Choisit SA_{ipsec1} de SA_{ipsec2} en fonction de sa préférence et envoie $E_{KAB}\{ SA_{ipsec2}, H(SA_{ipsec2})\}$ à l'initiateur. Si il n'accepte aucun algorithme existant dans SA_{ipsec1}, il peut rejeter la liste complète des SA et il renvoie une erreur dans le deuxième message à l'initiateur.

5.1.3. Analyse de la sécurité

5.1.3.1. Analyse théorique

Les propriétés de sécurité assurées par notre protocole:

- Known-Key Security: dans notre protocole, vu que K est calculé par deux nombres aléatoires (X_a, X_b) de sorte que chaque négociation complète devrait se traduire par une clé de session partagée unique. Par conséquent, le compromis d'une clé de session partagée ne doit pas compromettre les clés dans les autres sessions.

- La résilience contre l'attaque par rejeu: notre protocole peut résister à l'attaque par rejeu, parce que N1 est une valeur aléatoire et permet d'assure que la réponse n'a pas été rejouée par un adversaire.

- L'efficacité: notre protocole IKE ne nécessite qu'une seule phase, qui se compose de trois messages d'échange allers-retours. Les quatre premiers messages sont utilisés pour établir SA-IKE et les deux derniers messages, qui sont sous la protection de la clé de session partagée, sont utilisés pour établir le SA-IPSec.

- La résilience à l'attaque l'homme du milieu (man-in-the-middle): dans notre protocole, N1 est l'information secrète entre l'initiateur et le répondeur, l'utilisation de N1 est efficace pour authentifier les deux parties et la protection contre l'attaque de l'homme du milieu.

- La résilience de Control Key: puisque la clé est calculée telle que $K_{AB} = Y_b^{X_a} \bmod q$, aucune entité n'est capable de forcer la clé de session partagée comme une valeur présélectionnée.

- La défense contre l'attaque DoS: deux types de paquets d'inondation doivent être considérés: Msg3 et Msg5. Selon notre protocole, un message forgé de « Msg3 » serait totalement une cause pour que le répondeur exécute deux fois la fonction de hachage, une fonction XOR. D'autre part, un message forgé « Msg5 » provoquerait totalement le répondeur pour faire une fois la fonction de hachage et un chiffrement symétrique. Par conséquent, toutes ces opérations sont simples et pourraient être accomplies rapidement. Donc, une attaque DoS ne peut pas empêcher le service du répondeur, à moins que cette attaque reste en fonctionnement pour une période assez longue.

5.1.3.2. Vérification par analyse formelle des propriétés de sécurité du protocole proposé

Nous avons utilisé l'outil AVISPA (the automatic validation of the protocols and applications Internet sensitive to the security). Il fournit un langage formel modulaire et expressif pour spécifier les protocoles et leurs propriétés de sécurité. Il intègre différents back-end qui mettent en œuvre une variété de techniques d'analyse automatique de la machine.

L'outil AVISPA utilise le modèle d'intrusion Dolev-Yao où l'attaquant peut espionner tous les messages transmis, usurper l'identité d'une entité légitime (l'attaque par usurpation d'identité) et modifier ou injecter des messages, mais il considère que la cryptographie est parfaite, à savoir l'attaquant ne peut pas briser la cryptographie. Le cadre AVISPA est représenté dans la figure 5.2 ci-dessous.

La première étape dans l'utilisation de l'outil est de présenter le protocole analysé dans un langage spécial appelé HLPSL (High Level Protocol Specification Language). La spécification de HLPSL du protocole est traduite dans la langue de niveau inférieur appelé IF (Intermediate Format). Cette traduction est effectuée par le traducteur appelé HLPSL2IF. Cette étape est totalement transparente à l'utilisateur.

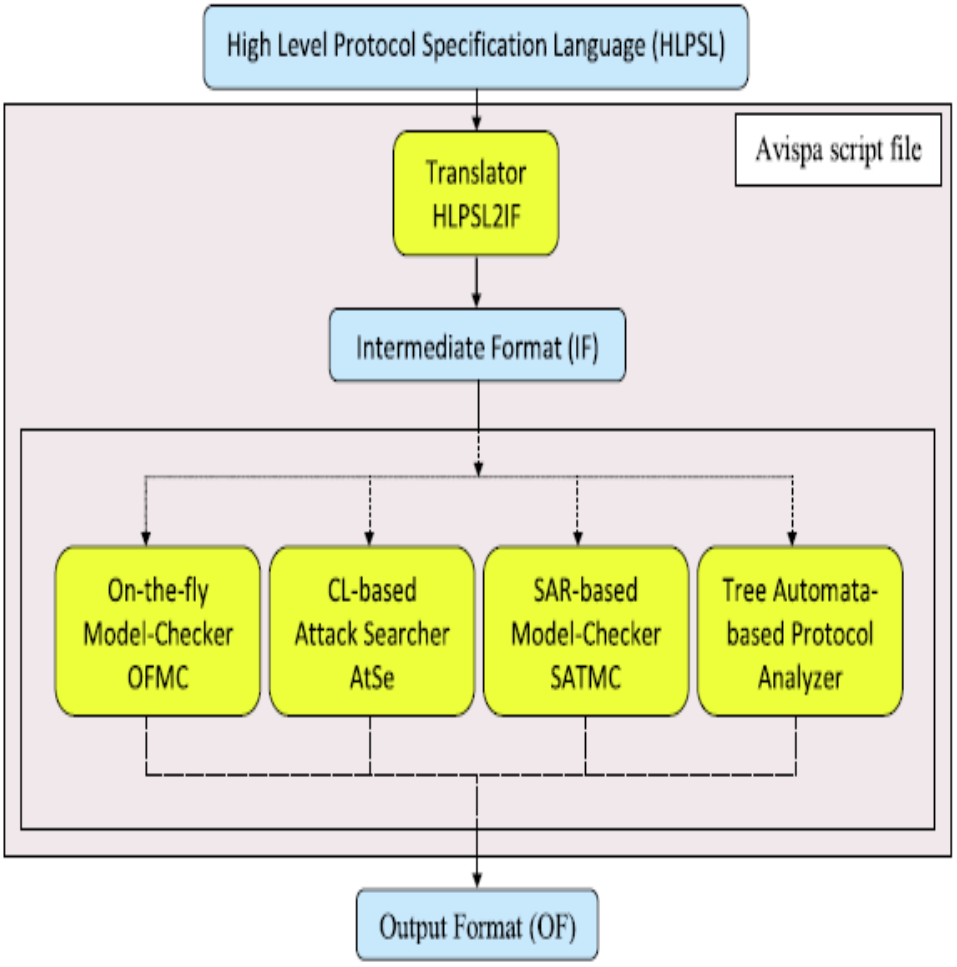

Figure 5.2 L'architecture d'AVISPA (Farash et al., 2013)

La spécification IF du protocole est utilisée comme une entrée aux quatres différents back-end: On-the-fly Model-Checker (OFMC), CL-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC) et Tree-Automata-based Protocol Analyzer (TA4SP).

Ces back-end effectuent l'analyse et le formatage de sortie qui contient les résultats. Ces résultats indiquent s'il y a des problèmes dans le protocole ou non (AVISPAv1.1., 2006; Glouche et al., 2006; Farash et al., 2013).

5.1.3.2.1. La spécification du protocole IKE

Le protocole proposé a été modélisé dans un langage appelé HLPSL et écrit dans un fichier avec l'extension hlpsl (IKE.hlpsl). Ce language est basée sur les rôles. Il ya trois rôles fondamentaux « Alice », « Bob » et « AS ». Nous ne présentons que l'un des rôles de base « Bob » qui est indiqué dans la figure 5.3.

Après avoir défini les rôles de base, nous devons définir les rôles composés qui décrivent les sessions du protocole. Ensuite, un rôle de haut niveau est défini. Ce rôle contient les constantes globales, la connaissance initiale des intrus et la composition de plusieurs sessions.

Enfin, la section objectif est utilisée pour préciser les objectifs de sécurité, comme étant les propriétés qui permettent à l'outil AVISPA de rechercher les attaques.

Les évènements « Witness » et « Request » sont les propriétés d'authentification alors que l'évènement « Secrecy » est utilisé pour vérifier le secret partagé entre les agents « Alice » et « Bob».

La validation de la représentation du protocole modélisée a été réalisée en utilisant un outil appelé SPAN. Après cette étape, ce protocole est exécuté contre l'intrus modélisé afin de vérifier les exigences de sécurités souhaitées pour vérifier les forces et les faiblesses en utilisant d'outil AVISPA (OFMC).

```
role bob (…) played_by B def=
local const   init  State := 0 transition
 State = 0 /\ Rcv(SA1') =|>
      State':= 1 /\ SA2':=new()
                  /\Snd(SA2)
 State = 1  /\ Rcv(xor(N1'.pb)) =|>  State':=
2
  /\ witness(B,S,sk2,N1')
   /\ N2':= xor(xor(N1',pb),pb) …….
 State = 4 /\ Rcv({SA3'.H(SA3')}_KAB)   =|>
State':=5   /\ Snd({SA3.H(SA3)}_KAB)
             /\ secret(KAB,sec_KA,{A,B})
end role
```

Figure 5.3 Le rôle de Bob

5.1.3.2.2. Analyse des résultats

Nous avons choisi la back-end OFMC du cadre AVISPA afin de vérifier la sécurité de notre protocole:

- L'attaque l'homme du milieu (man-in-the-middle): la description du rôle de l'environnement est donnée ci-dessous. Cette description permet de détecter l'attaque de l'homme du milieu si elle existe.

```
role environment() def= intruder_knowledge = {a,b,s,i}
  composition
    session (a,b,s,g,snd,rcv)
/\ session (a,i,s,g,snd,rcv)
/\ session (i,b,s,g,snd,rcv)
end role
```

Figure 5.4 Le rôle de l'environnement 1

Les résultats (Figure 5.5) indiquent que notre protocole est sûr contre les attaques «man-in-the-middle ».

- La résilience contre l'attaque par rejeu: lorsque vous utilisez l'option-sessco, OFMC, elle effectuera d'abord une recherche avec un intrus passif pour vérifier si les agents honnêtes peuvent effectuer le protocole. Les résultats montrent que notre protocole peut résister à une attaque par rejeu.

Ce travaille a était accepté par la conférence ICMASM'2013, puis sélectionné et publié par le journal « International Journal of Information Security Research » Vol.4, No.3.

Notre protocole possède plusieurs avantages, ce qui fait de lui le meilleur protocole par rapport aux protocoles existants dans la littérature. Parmi ces avantages, nous citons: son fonctionnement est basé sur une phase (Vs. deux phases dans le protocole standard d'IKE); il résiste aux différents types d'attaques tels que (DoS, l'homme du milieu, par rejeu).

Les deux principaux inconvénients de notre protocole IKE ainsi que les protocoles existants dans la littérature sont: la complexité élevée, car ils ont besoin de beaucoup de calculs modulaires pour créer la clé secrète; ils utilisent une taille de clé assez grande afin d'assurer un niveau de sécurité élevé.

De plus, la plupart des articles étudiés sont concentrés sur la résistance contre les attaques: DoS et l'homme du milieu, mais ils négligent les autres types d'attaques telles que l'attaque par rejeu, par réflexion et modification. Ces derniers causent la perte de la dimension de la sécurité comme l'authentification, la non-répudiation, l'intégrité et la confidentialité. Par conséquent, la sécurité de l'information sera endommagée. Notre première proposition résiste à l'attaque par rejeu.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\SPAN\testsuite\results\IKE.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 2.37s
  visitedNodes: 1624 nodes
  depth: 11 plies
```

Figure 5.5 Les résultats indiqués par le back-end OFMC

Cette synthèse nous a permis de proposer un nouveau protocole IKE basé sur la cryptographie à courbe elliptique qui vérifie plusieurs dimensions de sécurité (authentification, non-répudiation, l'intégrité et la confidentialité) avec une clé de petite taille et de faible complexité, ce qui rend notre protocole IKE efficace et sécurisé.

5.2. La deuxième proposition du protocole IKE

La deuxième proposition vise à construire un protocole IKE sécurisé avec une charge de calcul faible pour améliorer la sécurité et rendre la phase d'initialisation d'IPSec plus léger. Contrairement aux travaux connexes, le protocole proposé peut résister aux différents types d'attaques telles que la modification, par réflexion, par rejeu, DoS et l'homme du milieu (man-in-the-middle) avec moins de complexité de calcul (voir la section 5.4).

5.2.1 Le nouveau IKE basé sur ECC

Dans ce travail, nous avons utilisé le protocole d'authentification conforme à la courbe elliptique proposée dans (Zeyad et al., 2011) pour construire un protocole IKE efficace et sécurisé. Notre protocole proposé est composé de cinq messages. Les trois premiers messages sont utilisés pour établir SA-IKE, et les deux derniers messages sont utilisés pour établir SA-IPSec. La figure 5.6 illustre le cadre du protocole IPSec via notre protocole IKE proposé. Pour illustrer clairement le fonctionnement de notre protocole, les notations utilisées par ce protocole sont présentées dans la sous-section 5.3.1.1. L'architecture du protocole proposé est décrite en détail dans la sous-section 5.3.1.2.

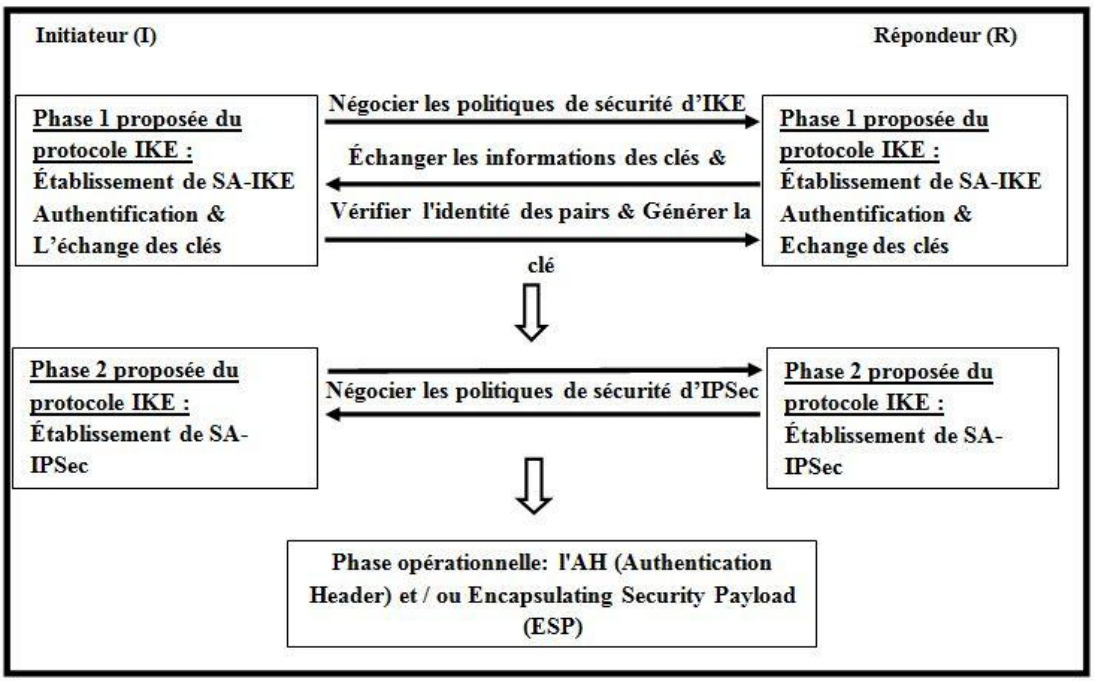

Figure 5.6 Le cadre d'IPSec en utilisant notre protocole IKE proposé

5.2.1.1. Notations

Nous avons utilisé les notations suivantes pour décrire notre protocole IKE:

- I : l'initiateur;
- R : le répondeur;
- ID_I : l'identité de l'initiateur;
- ID_R: l'identité du répondeur;
- P : générateur de point de ECC;
- ⟨P⟩ : un sous-groupe cyclique ⟨P⟩ de $E(F_q)$;
- w_I, w_R : les clés privées statiques de I et R, où w_I, $w_R \in [1; n - 1]$;
- W_I, W_R : les clés publiques statiques d'I et R, où $W_I = w_I .P$ et $W_R = w_R .P$;
- r_I, r_R : les clés privées éphémères de I et R;
- R_I, R_R : les clés publiques éphémères d'I et R;
- SA_{I1} : l'association de sécurité proposée d'IKE;
- SA_{I2} : l'association de sécurité sélectionnée d'IKE;
- H1, H2 : les fonctions de hachage H;
- SA_{ipsec1}: l'association de sécurité proposée d'IPSec;
- SA_{ipsec2}: l'association de sécurité sélectionnée d'IPSec;
- K : la clé de session éphémère calculée par l'initiateur et le répondeur;
- K_{IR} : la clé de session obtenue par l'initiateur et le répondeur;

- ❖ || : enchaînement;
- ❖ ⊕ : XOR ;
- ❖ E_{KIR} (.): cryptage en utilisant un système de chiffrement symétrique par la clé K_{IR}.

5.2.1.2. La description du protocole proposé

Comme il est représenté dans la figure 5.7, notre protocole est composé de cinq messages. Les trois premiers messages sont utilisés pour établir le SA-IKE, l'authentification mutuellement et l'échange des clés secrètes. Les deux derniers messages sont utilisés pour établir SA-IPSec.

- ❖ Étape 1: L'initiateur ⟶ Le répondeur : ID_I, ID_R, SA_{i1}, R_I

  Dans un premier temps, l'initiateur sélectionne un nombre aléatoire $r_I \in [1, n-1]$ et il calcule $R_I = H2 (r_I \| w_I) P$. Puis, il envoie au répondeur ID_I, ID_R, R_I et une liste de propositions de paramètres confidentiels SA_{i1}.

- ❖ Étape 2: Le répondeur ⟶ L'initiateur : ID_I, ID_R, SA_{i2}, R_I, R_R, Z_R

  Lors de la réception du message d'initiateur, le répondeur effectue les opérations suivantes:
  - ➢ Le répondeur sélectionne SA_{i2} de SA_{i1} selon sa préférence. Si le répondeur n'accepte aucun algorithme existant dans SA_{i1}, il peut rejeter la liste complète des SA et il renvoie une erreur dans le deuxième message à l'initiateur;
  - ➢ Il vérifie $R_I \in P^*$ en utilisant une validation de la clé R_I. Si $R_I \notin P^*$ le répondeur termine l'exécution du protocole avec une sortie en échec;
  - ➢ Il sélectionne un nombre aléatoire $r_R \in [1, n-1]$, et il calcule $R_R = H2 (r_R \| w_R) P$;
  - ➢ Il calcule: $K = H1 (r_R \| w_R) R_I$, $e_R = H2 (ID_I \| ID_R \| SA_{i1} \| x_{RI} \| x_{RR} \| x_K)$, $c_R = H1 (r_R \| w_R) + w_R \bmod q$ et $Z_R = c_R \oplus e_R$, où x_{RI} désigne la coordonnée x de R_I, x_{RR} désigne la coordonnée x de R_R et x_K désigne la coordonnée x de K;
  - ➢ Il envoie ID_I, ID_R, SA_{i2}, R_I, R_R, Z_R à l'initiateur.

- Étape 3: L'initiateur ⟶ Le répondeur : ID_I, ID_R, SA_{i2}, R_I, R_R, Z_I, Z_R

  À la réception du message du répondeur, l'initiateur effectue les opérations suivantes:

  - Il compare les paramètres reçus ID_I, ID_R, SA_{i2}, R_I, R_R, Z_R à ses propres paramètres ID_I, ID_R, R_I, $SA_{i2} \in SA_{i1}$?. Si ils ne correspondent pas, alors l'initiateur termine l'exécution du protocole avec une sortie en échec;
  - Il vérifie si $R_R \in P^*$ en utilisant la clé de validation de R_R. Si $R_R \notin P^*$, l'initiateur termine l'exécution du protocole avec une sortie en échec;
  - Il calcule: $K= H1 (r_I \| w_I) R_R$, $é_R=H2 (ID_I \| ID_R \| SA_{i1} \| x_{RI} \| x_{RR} \| x_K)$, $c_R= Z_R \oplus é_R$ donc $c_R= c_R \oplus e_R \oplus é_R$;
  - Il vérifie $W_R= c_R P-R_R$? Si la vérification échoue, l'initiateur termine l'exécution; sinon, il engage la procédure de calcule suivante: $e_I=H2 (ID_I \| ID_R \| SA_{i2} \| y_{RR} \| y_{RI} \| x_K)$, $c_R= H1 (r_I \| w_I) + w_I \bmod q$ et $Z_I= c_I \oplus e_I$, où y_{RI} désigne la coordonnée y de R_I, où y_{RR} désigne la coordonnée y de R_R, x_K désigne la coordonnée x de K;
  - Il envoie ID_I, ID_R, SA_{i2}, R_I, R_R, Z_I, Z_R au répondeur.

- Étape 4: À la réception du message d'initiateur, le répondeur effectue les opérations suivantes:

  - Il compare les paramètres reçus ID_I, ID_R, SA_{i2}, R_I, R_R, Z_I, Z_R avec ses propres paramètres ID_I, ID_R, SA_{i2}, R_I, R_R, Z_R. Si ils ne correspondent pas alors le répondeur termine l'exécution du protocole avec une sortie en échec;
  - Il calcule: $é_I=H2 (ID_I \| ID_R \| SA_{i1} \| y_{RR} \| y_{RI} \| x_K)$, $c_I= Z_I \oplus é_I$ donc $c_I= c_I \oplus e_I \oplus é_I$;
  - Il vérifie $W_I= c_I P-R_I$? Si la vérification échoue, il termine l'exécution. Sinon, le répondeur assure l'identité de l'initiateur.

Une fois l'authentification réussie entre l'initiateur et le répondeur, ils s'accordent sur une clé de session commune qui est calculée comme suit:

$K_{IR}= (ID_I \| ID_R \| SA_{i2} \| x_{RI} \| x_{RR} \| c_I \| c_R \| x_K)$.

❖ Étape 5 : L'initiateur ⟶ Le répondeur : E_{KIR} {ID_I‖ ID_R‖ $SA_{proposal\text{-}ipsec}$}

L'initiateur crypte le ID_I‖ ID_R‖ $SA_{proposal\text{-}ipsec}$ en utilisant la clé de cryptage K_{IR} généré précédemment et il l'envoie au répondeur.

❖ Étape 6: Le répondeur ⟶ L'initiateur : E_{KIR} {ID_I‖ ID_R‖ $SA_{selected\text{-}ipsec}$}

Lors de la réception d'un message de l'initiateur, le répondeur effectue les opérations suivantes:

- Il décrypte le message crypté reçu par K_{IR};
- Il sélectionne $SA_{selected\text{-}ipsec}$ de $SA_{proposal\text{-}ipsec}$, selon sa préférence. Si le répondeur n'accepte aucun algorithme existant dans le SA proposé, il peut rejeter la liste complète de $SA_{proposal\text{-}ipsec}$ et il renvoie une erreur dans le deuxième message à l'initiateur.

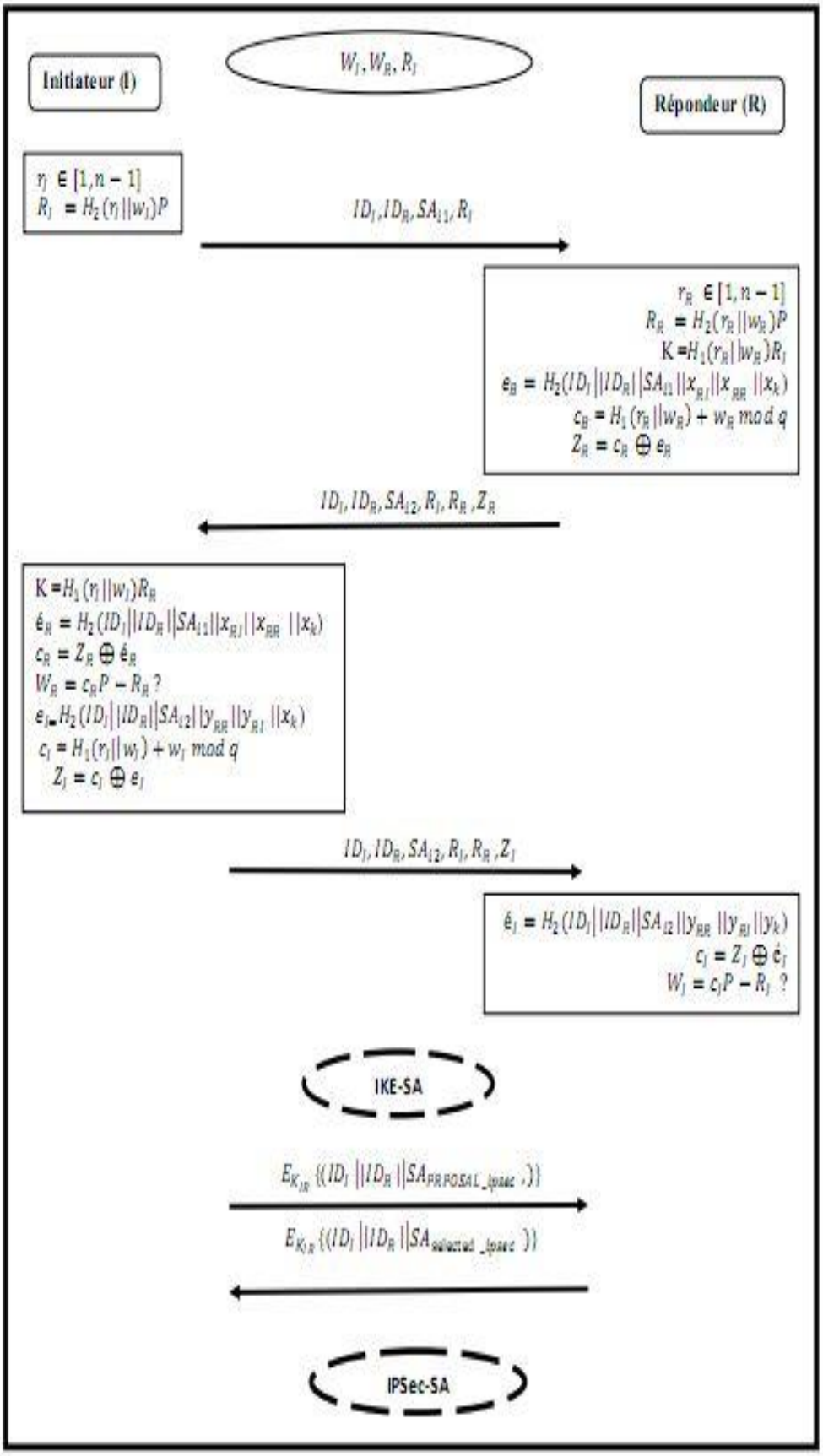

Figure 5.7 Notre protocole IKE

5.2.1.3. Evaluation de la sécurité du protocole proposé

Nous présentons dans ce qui suit l'analyse de la sécurité et de la vérification formelle de notre protocole IKE, qu'elles montrent que notre protocole peut surmonter les faiblesses mentionnées précédemment.

5.2.1.3.1. Analyse théorique

Le protocole proposé a les propriétés suivantes:

- Perfect Forward Secrecy: le compromis des clés à long terme ne devrait pas conduire à la compromission des clés des sessions précédentes. Dans notre protocole, les clés privées r_I, w_I, r_R, w_R sont choisies au hasard. Par conséquent, le compromis K_{IR} ne peut être utilisé pour dériver d'autres clés de session.

- L'attaque par Réflexion: est une méthode qui sert à attaquer un système d'authentification défi-réponse. Elle utilise le même protocole dans les deux directions. Dans notre protocole, l'initiateur est en mesure de vérifier l'identité du répondeur dans le second message (R_R, Z_R). De la même manière, le répondeur peut vérifier l'identité de l'initiateur dans le troisième message (Z_I) (Z_I). (e.g., $c_I = Z_I \oplus é_I$ donc $c_I = c_I \oplus e_I \oplus é_I = c_I \oplus 0$, $W_I = c_I P\text{-}R_I$?).

- L'attaque par rejeu: notre protocole peut résister à l'attaque par rejeu. Nous supposons que l'adversaire « C » espionne la conversation entre l'initiateur «I» et le répondeur « R », lorsque l'échange est fini entre «I» et «R». Cet intrus «C» se connecte à «R». « C » envoie à « R » R_I lit à partir de la dernière session. Ensuite, le répondeur envoie R_R, Z_R à l'adversaire. Mais dans la troisième étape, l'adversaire est attrapée, car il ne peut produire le Z_I qui correspond à R_R.

- Known-key security: l'exécution de notre protocole produit une clé de session unique, puisque les paramètres éphémères dans cette dernière sont aléatoires et indépendants des autres clés de session.

- Défense d'attaque DoS: dans notre protocole, il ya trois types de paquets d'inondations: Msg1, Msg3 et Msg4. Pour le premier type, une forgée Msg1 poussera le répondeur à passer du temps pour une fonction de hachage, addition,

multiplication. Pour le second type, une forgée Msg3 poussera le répondeur à exécuter une fonction de hachage, la multiplication. Pour le troisième type, une forgée Msg4 poussera le répondeur à exécuter une seule fois le chiffrement symétrique. Par conséquent, toutes ces opérations sont simples et pourraient être accomplies rapidement. Donc, une attaque DoS ne peut empêcher le service du répondeur, à moins que cette attaque reste en opération pour une période assez longue.

- La Clé de contrôle (Control Key): avec notre protocole, aucune partie n'est capable de forcer la clé de la session partagée à une valeur présélectionnée.

- Key compromise impersonation : compromission de la clé privée éphémère r_I ou la clé privée statique w_I permet à un adversaire de prendre l'identité I, mais il ne permet pas l'adversaire de prendre l'identité des autres entités pour I. Par conséquent, notre protocole a cette propriété.

- L'efficacité: notre protocole IKE possède les propriétés du système de cryptographie à courbe elliptique comme une clé de petite taille, donc nous avons moins d'espace pour le stockage des clés et une faible charge de calcul. En outre, notre protocole est composé de cinq messages. Les trois premiers messages sont utilisés pour établir SA-IKE et les deux derniers messages sont utilisés pour établir IPSec-SA.

- L'attaque de l'homme du milieu: supposons que l'adversaire espionne le canal de communication entre l'initiateur (I) et le répondeur (R). Il peut remplacer la demande d'authentification Z_I par Z_C. Cependant, l'attaque de l'homme du milieu ne peut pas réussir en raison de la vérification de W_I par le répondeur dans la troisième étape. Par conséquent, notre protocole peut résister à l'attaque de l'homme du milieu.

5.2.1.3.2. Vérification formelle

En plus de l'analyse théorique, nous fournissons une analyse formelle de notre protocole en utilisant l'outil AVISPA (AVISPA., 2006; Basu et al, 2012) (voir la

section 5.1.3.2) pour valider les différentes propriétés de sécurité assurées par notre protocole. Nous présentons le rôle de Bob dans la figure 5.8.

```
role bob ()

const ok,reject :message, sec_b_KAB:protocol_id

init State := 0

transition

1.    State=0  /\ RCV_I(IID'.RID'.SAI1'.RIi')/\

not(in(RIi',S))   =|>  State':= 1 /\ SND_I(reject)

2. State=0  /\ RCV_I(IID'.RID'.SAI1'.RIi')/\ (in(RIi',S))=|>

State':=2 /\  Rr':=new()  /\  Wr':=new() /\RR':=H2(Rr'.Wr').P
/\K':=H1(Rr'.Wr').RIi'
/\Er':=H2(IID'.RID'.SAI1'.F1(RIi').F2(RR').F3(K'))/\Cr':=H1(Rr
'.Wr').Wr  /\Zr':=xor(Cr',Er') /\ SND_I
(IID'.RID'.SAI1'.RIi'.RR'.Zr') /\
witness(A,B,sk2,H1(Rr'.Wr').RIi')

3.State =2 /\  RCV_I(reject)=|> State' :=3

4.State =2 /\  RCV_I(Idi'.Idr'.SAI1'.RIi'.RR'.Zi')=|>

State' :=4 /\ Ei':= H2(Idi'.Idr'.SAI1'.F4(RR').F5(RIi').F3(K))

/\ Ci1':= xor(Zi',Ei')  /\Ci2':=Ci1'.P4 /\
Ci3':=xor(Ci2',RIi')

5. State =4 /\ (Ci3 = WR)=|> State' :=5 /\ SND_I(ok)

6. State =5 /\RCV_I({IID.RID.SAIPSEC'}_ KAB')=|> State':=6

/\KAB':=H(IID.RID. SAI1. F1(RIi). F2(RRr). Ci1. Cr.F3(K))

/\ SND_I ({IID.RID.SAIPSEC'}_ KAB')

/\ witness(A,B,kab2,H(IID.RID.F1(RIi).F2(RRr).Ci1.Cr.F3(K)))

/\secret(KAB,sec_b_KAB,{A,B}) /\ request (A,B,kab1,KAB)

end role
```

Figure 5.8 Le rôle de Bob

- L’analyse des résultats: nous choisissons le back-end OFMC du cadre AVISPA afin de vérifier la sécurité de notre protocole:

  - L'attaque de l'homme du milieu: la spécification en langage HLPSL pour le rôle de l'environnement 1 est donnée dans la figure 5.9. Ce rôle

d'environnement est utilisé pour détecter l'attaque de l'homme du milieu. Les résultats présentés dans la figure 5.11 indiquent que notre protocole peut résister à l'attaque de l'homme du milieu.

```
role environement ()def=

intruder_knowledge ={a,b,I,g,p,h,h1,h2}

composition

session(a,b,g,p,h,h1,h2,wr,wi,ri,rr,snd,rcv)/\
session(a,i,g,p,h,h1,h2,wr,wi,ri,rr,snd,rcv)/\

session(i,b,g,p,h,h1,h2,wr,wi,ri,rr,snd,rcv)

end role
```

Figure 5.9 Le rôle de l'environnement 1

- L'attaque par rejeu: la spécification en langage HLPSL pour le rôle de l'environnement 2 est donnée dans la figure 5.9. Ce rôle d'environnement est utilisé pour détecter l'attaque par rejeu. Les résultats présentés dans la figure 5. 11 indiquent que notre protocole peut résister à l'attaque par rejeu.

```
role environement ()def=

intruder_knowledge = {a,b,i,g,p,h,h1,h2}

composition

session(a,b,g,p,h,h1,h2,wr,wi,ri,rr,snd,rcv)/\

session(a,b,g,p,h,h1,h2,wr,wi,ri,rr,snd,rcv)

end role
```

Figure 5.10 Le rôle de l'environnement 2

- Delov-Yao Model Check: à la fin, la profondeur que nous avons choisi pour la recherche est de six et la sortie du modèle de vérification des résultats est présenté dans la figure 5.11. Comme le montre la figure, il y a un total de 116 nœuds qui ont été fouillés en 0,32 s. D'après ces résultats, nous pouvons conclure que le protocole proposé est sécurisé sous le test d'AVISPA en utilisant le back-end OFMC avec un nombre borné de sessions.

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
C:\SPAN\testsuite\results\IKE.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.32s
visitedNodes: 116 nodes
depth: 6 plies
```

Figure 5.11 Les résultats d'analyse formelle en utilisant le back-end OFMC

Cette proposition a été acceptée et publiée par le journal « International Journal of Communication Networks and Distributed Systems » (Ahmim et al., 2015). Notre protocole satisfait les exigences de sécurité suivantes: authentification mutuelle entre les deux parties, clé de session avec la propriété Perfect Forward Secrecy, la confidentialité et l'intégrité des données. En outre, notre protocole contient cinq messages. Les trois premiers messages sont utilisés pour établir SA-IKE, et les deux derniers messages sont utilisés pour établir SA-IPSec. En outre, il possède les propriétés du système de cryptographie à courbe elliptique comme une taille de clé courte, faible charge de calcul. Ainsi, le protocole proposé est efficace en termes de

complexité de calcul. Enfin, nous avons démontré que le protocole proposé peut résister à divers types d'attaque par la vérification formelle en utilisant l'outil AVISPA, ce qui rend notre protocole plus efficace dans les réseaux sans fil.

Afin de rendre ce protocole plus efficace et plus sécurisé, on a proposé un nouveau protocole appelé, EIAKEP, « An efficient internet authenticated-key exchange protocol ». Ce protocole est présenté dans la section suivante.

5.3. Le protocole EIAKEP « An efficient internet authenticated-key exchange protocol »

L'EIAKEP est composée de quatre messages d'échange. Dans les deux premiers messages l'initiateur et répondeur créent SA-IKE, s'authentifient mutuellement et ils échangent leurs clés secrètes. Les deux derniers messages sont utilisés pour établir SA-IPSec.

Contrairement aux travaux connexes, ainsi que nos deux premières contributions, le protocole proposé peut satisfait à toutes les propriétés de sécurité du protocole de gestion de clé et il peut résister aux différents types d'attaques telles que la modification, la réflexion, la relecture, DoS et l'homme du milieu avec moins de complexité de calcul et moins d'échange.

5.3.1 Notations

Nous utilisons les notations énumérées ci-dessus pour décrire l'EIAKEP.

- ID_i : l'identité de l'initiateur;
- ID_r : l'identité du répondeur;
- P: une génération de point ECC d'ordre premier dans E (F_q);
- t_i, t_r: des clés privées statiques d'I et R, où t_i, $t_r \in [1, n-1]$;
- T_i, T_r : des clés publiques statiques d'I et R;
- SA_{IKE1} : une liste de propositions cryptographiques de l'initiateur (l'association de sécurité proposé de IKE);
- SA_{IKE2} : une liste des protocoles cryptographiques sélectionnés par le répondeur parmi la liste envoyée par l'initiateur (l'association de sécurité choisie);
- u_i, u_r: les clés privées d'I et de R;

- U_i, U_r : des clés publiques de l'I et le R, où: Ui = u_i.P, Ur = u_r.P ;
- H : fonctions de hachage;
- SA_{ipsec1} : une liste de propositions cryptographiques de l'initiateur (l'association de sécurité proposé d'IPSec);
- SA_{ipsec2} : une liste des protocoles cryptographiques sélectionnés par le répondeur de la liste envoyée par l'initiateur (l'association de sécurité choisie);
- K : la clé de session éphémère calculée par l'initiateur et le répondeur;
- K_{ir} : la clé de session obtenue par l'initiateur et le répondeur;
- E_{Kir} (.) : le cryptage en utilisant un système de chiffrement symétrique avec la clé $K_{ir.}$

5.3.2 La description du protocole

L'EIAKEP basé sur ECDH entre I et R est représenté dans la figure 5.12. Ce protocole utilise la conception de base de la signature El-Gamal avec des modifications qui garantissent les propriétés de sécurité et l'efficacité du protocole de gestion des clés. L'EIAKEP comporte quatre étapes:

- Étape 1: L'initiateur → le répondeur: $T_{i,}$ SA_{IKE1}

Dans un premier temps, l'initiateur sélectionne un nombre aléatoire $t_i \in [1, n-1]$, il calcule T_i= H ($t_i \| u_i$).P, puis il envoie au répondeur $T_{i,}$ SA_{IKE1}.

- Étape 2: Le répondeur → L'initiateur: $SA_{IKE2,}$ $T_{r,}$ $U_{r,}$ $S_{r,}$ β

Lors de la réception d'un message de l'initiateur, le répondeur effectue les opérations suivantes:

- Le répondeur sélectionne un SA_{IKE2} de SA_{IKE1} selon sa préférence. Si le répondeur n'est pas d'accord pour une SA, il peut rejeter la totalité de la liste de la proposition SA et renvoie une erreur dans le deuxième message.
- Il sélectionne un nombre aléatoire $t_r \in [1, n-1]$ et calcule T_r= H ($t_r \| u_r$).P
- Il calcule: K=H ($t_r \| u_r$).T_i, S_r=K.[H(T_r)-H($t_r \| u_r$)(X_{ur})],
  β= H($ID_i \| ID_r \| SA_{IKE1} \| X_K$), où X_K indique la coordonnée x de K.
- Il envoie $SA_{IKE2,}$ $T_{r,}$ $U_{r,}$ $S_{r,}$ β à l'initiateur.

- Étape 3 : L'initiateur → Le répondeur: E_{Kir}{(U_i, $ID_{r,}$ SA_{ipsec1}, α, S_i}

À la réception du message du répondeur, l'initiateur effectue les opérations suivantes:

- Calcule: $K=H\ (t_i \| u_i)T_r$, $V1=[X_{ur}.T_r.K+S_r.P]$, $V2=H(T_r).K.P$. Ensuite, il vérifie si V1 = V2. Si la vérification échoue, l'initiateur termine l'exécution; sinon, il calcule $\beta' = (ID_i \| ID_r \| SA_{IKE1} \| X_K)$ et il vérifie si $\beta = \beta'$. Si la vérification échoue, l'initiateur termine l'exécution; sinon, il calcule $S_i=K.[H(T_i)-H(t_i \| u_i).(X_{ui})]$, $\alpha = H(ID_i \| ID_r \| SA_{IKE2} \| X_K)$ et $K_{ir}= H(ID_i \| ID_r \| X_{Ti} \| X_{Tr} \| X_K)$;
- Il chiffre U_i, ID_r, SA_{ipsec1}, α, S_i par la clé K_{ir} ;
- Il envoie au répondeur $E_{Kir}\{U_i, ID_r, SA_{ipsec1}, \alpha, S_i\}$.

❖ Étape 4: Le répondeur → L'initiateur: $E_{Kir}\{ID_r, SA_{ipsec2}\}$

À la réception du message de l'initiateur, le répondeur effectue les opérations suivantes:

- Il calcule $K_{ir}= H\ (ID_i \| ID_r \| X_{Ti} \| X_{Tr} \| X_K)$;
- Il décrypte le message crypté reçu par K_{ir};
- Il calcule $V1= [X_{ui}.T_i.K+S_i.P]$, $V2=H(T_i).K.P$. Ensuite, il vérifie si V1 = V2. Si la vérification échoue, l'initiateur termine l'exécution; sinon, il calcule $\alpha'= H(ID_i \| ID_r \| SA_{IKE2} \| X_K)$, puis il vérifie si $\alpha' = \alpha$. Si la vérification échoue, le répondeur termine l'exécution; sinon, il sélectionne une SA_{ipsec2} de SA_{ipsec1} selon sa préférence.
- Il envoie $E_{Kir}\{ID_r, SA_{ipsec2}\}$ à l'initiateur.

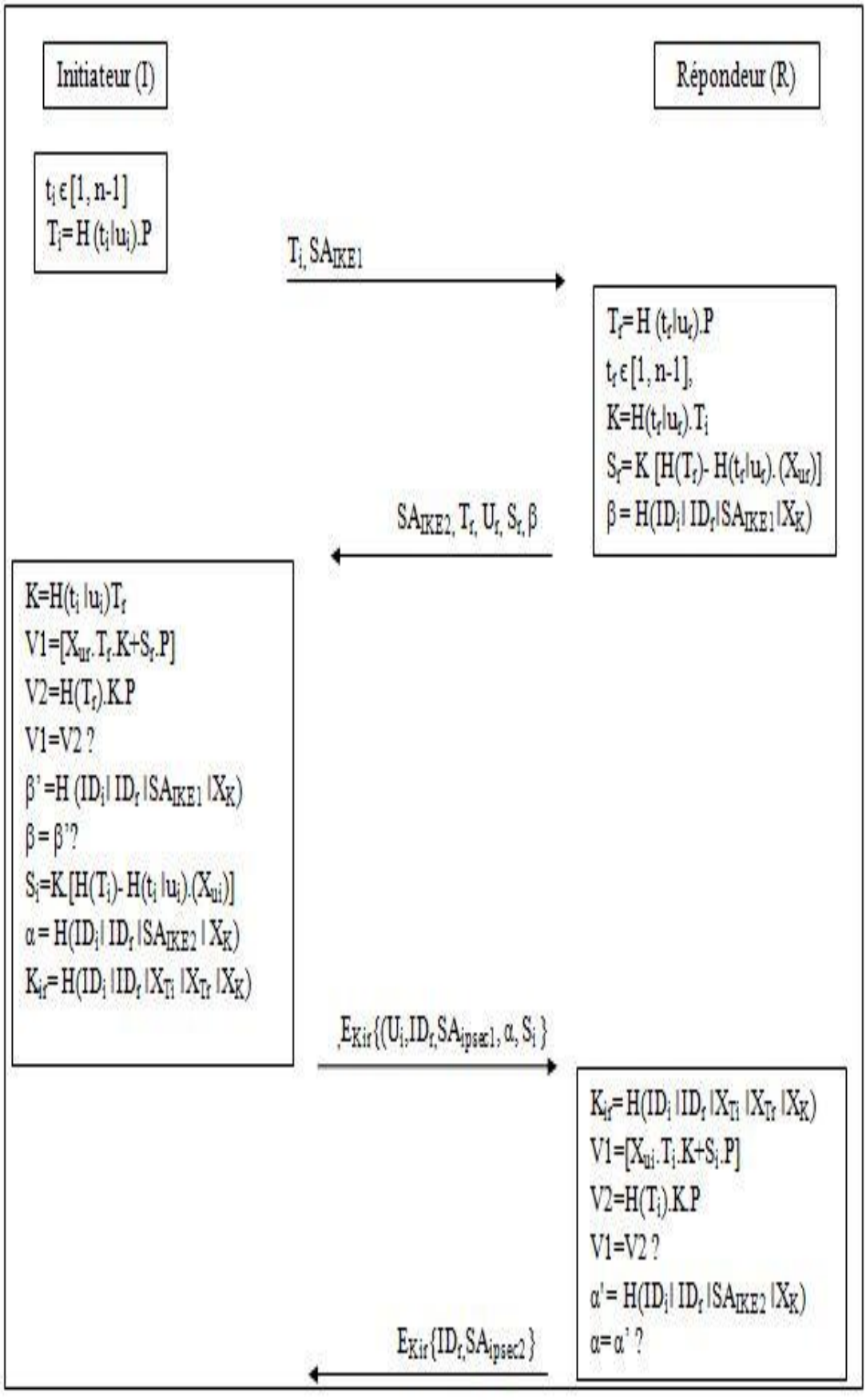

Figure 5.12 Le shéma EIAKEP proposé

5.3.3 Evaluation de la sécurité du protocole proposé

Nous présentons dans ce qui suit l'analyse de la sécurité et de la vérification formelle de notre protocole IKE, qui montrent que celui-ci peut surmonter les faiblesses mentionnées précédemment.

5.3.3.1 Analyse théorique

Le protocole proposé a les propriétés suivantes:

- Perfect Forward Secrecy: le compromis des clés à long terme ne devrait pas conduire à la compromission des clés des sessions précédentes. Dans notre protocole, les clés privées t_i, u_i, t_r, u_r sont choisis au hasard. Par conséquent, le compromis K_{ir} ne peut être utilisé pour dériver d'autres clés de session. Donc, l'EIAKEP satisfait la propriété Perfect Forward Secrecy.

- L'attaque par rejeu: notre protocole peut résister à l'attaque par rejeu. nous supposons que l'adversaire « C » espionne la conversation entre l'initiateur «I» et le répondeur « R », lorsque l'échange est fini entre «I» et «R». Cet intrus «C» se connecte à «R». « C » envoie à « R » T_i lu à partir de la dernière session. Ensuite, le répondeur envoie T_r, S_r, β à l'adversaire. Mais dans la troisième étape, l'adversaire est attrapée, car il ne peut produire la clé K.

- Known-key security: Dans le EIAKEP, les paramètres de la clé de session sont aléatoires et indépendants des autres clés de session. Par conséquent, le Known-key security est satisfait dans notre protocole.

- La défense d'attaque DoS: Dans le EIAKEP, il existe deux types de paquets d'inondations: Msg1 et Msg3. Pour le premier type, une forgée Msg1 poussera le répondeur à passer du temps pour la fonction de hachage et l'addition. Pour le second type, une forgée Msg3 poussera le répondeur d'exécuter une fois la fonction de hachage, la multiplication, le chiffrement et le déchiffrement symétriques. Par conséquent, toutes ces opérations sont simples et pourraient être accomplies rapidement. Donc, une attaque DoS ne peut empêcher le service de

répondeur, à moins que cette attaque reste en opération pour une période assez longue.

- La Clé de contrôle (Control Key): avec notre protocole, aucune partie n'est capable de forcer la clé de session partagée à une valeur pré-sélectionnée.

- Key compromise impersonation : compromission de la clé privée t_i ou u_i permet à un adversaire de prendre l'identité I, mais il ne permet pas à l'adversaire de prendre l'identité des autres entités pour I. Par conséquent, notre protocole a cette propriété.

- L'efficacité: l'EIAKEP utilise le système à courbe elliptique. Ce dernier utilise une clé de petite taille (128 bits) où l'utilisation des clés de longueur plus courte nécessite moins d'espace pour le stockage des clés, elle permet de gagner du temps lorsque les clés sont transmises et elle réduit les coûts de calcul arithmétique. Ces caractéristiques font du cryptosystème à courbe elliptique le meilleur choix pour améliorer la sécurité dans les réseaux de capture. Donc, notre protocole est efficace pour les réseaux de capture. Notre protocole n'a besoin que d'une phase, qui se compose de quatre échanges de messages. Les deux premiers messages sont utilisés pour établir le SA-IKE et les deux derniers messages sont utilisés pour établir le SA-IPSec.

- L'attaque de l'homme du milieu: supposons que l'adversaire espionne le canal de communication entre l'initiateur (I) et le répondeur (R). Il peut remplacer la demande d'authentification S_i avec S_c. Cependant, l'attaque de l'homme du milieu ne peut pas réussir en raison de la vérification de V1 par le répondeur dans la troisième étape. Par conséquent, l'EIAKEP peut résister à l'attaque de l'homme du milieu.

5.3.3.2 Vérification formelle

En plus de l'analyse théorique, nous fournissons une analyse formelle de notre protocole en utilisant les outils AVISPA (AVISPA., 2006; Basu et al, 2012) (voir la

section 5.1.3.2) pour valider les différentes propriétés de sécurité assurées par notre protocole. Nous présentons le rôle de Bob dans la figure 5.13.

```
rôle bob(B,A:agent,
           SND_A, RCV_A: channel (dy))
played_by B
def=
  local State: nat,
   …….
const sec_b_Kir : protocol_id
 init State := 1
 transition

1. State = 1  /\ RCV_A(SAIKE1'.Ti')  =|>
   State':= 3  /\ SAIKE2':= new()
                  /\ Uur':= new()  /\ Ttr':= new()
                  /\ IDi':= new() /\ IDr':= new()
                  /\ Ur':= F1(Uur',P)
                  /\ Eer':= (Uur'.Ttr')
                  ……..
                  /\ SND_A (SAIKE2'.Tr'.Ur'.Sr'.Wr')
                  /\ witness(B,A,sk2,K')

2. State=3 /\ RCV_A{Ui'.IDr'.SAIPSEC1'. Wi',Si'}_ Kir')  =|>
                State':=5
               /\ Kir':= H (IDi'.IDr'. F5(Ti').F5(Tr').F5(K'))
               /\ Y1':=F5(Ur')
               /\ V2':=F2(H1(Tr'),K')

3. State=5 /\ (V1=V2)=|> State':=6
           /\ SND_A  ({IDr'.SAIPSEC2'}_ Kir')
           /\secret(Kir,sec_b_Kir,{A,B})
           /\ request (A,B,kir2,Kir)

end rôle
```

Figure 5.13 Le rôle de Bob

Le protocole proposé est analysé dans le back-end OFMC, et le résultat est représenté dans la figure 5.14. D'après ce résultat, l'EIAKEP peut résister aux attaques passives et actives.

Notre protocole satisfait les exigences de sécurité suivantes: l'authentification mutuelle entre les deux parties, la clé de session avec la propriété Perfect Forward Secrecy, la confidentialité et l'intégrité des données. En outre, notre protocole n'a besoin que d'une phase, qui se compose de quatre échanges de messages. Les deux premiers messages sont utilisés pour établir SA-IKE et les deux derniers messages sont utilisés pour établir SA-IPSec. En outre, il possède les propriétés du système de cryptographie à courbe elliptique comme une taille de clé courte, une faible charge de calcul, et les exigences de bande passante. Ainsi, le protocole proposé est plus

efficace en termes de complexité de calculée. Enfin, nous avons démontré que le protocole proposé peut résister à divers types d'attaque par la vérification formelle en utilisant les outils AVISPA, ce qui le rend plus efficace dans les réseaux de capteurs.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\SPAN\testsuite\results\EIAKEP.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.73s
  visitedNodes: 208 nodes
  depth: 6 plies
```

Figure 5.14 Les résultats indiqués par le back-end OFMC

5.4. Etude comparative

Dans cette section, nous présentons une comparaison en termes de sécurité et de performance de nos protocoles ainsi que les protocoles existants dans la littérature.

Le tableau 5.1 représente une étude comparative où la métrique de sécurité utilisée pour mesurer la sécurité du protocole IKE est le nombre d'attaques.

La plupart des documents étudiés se concentrent sur la résistance contre les attaques DoS et l'homme du milieu, mais ils négligent les autres types d'attaques telles que : par rejeu (Replay), par réflexion, modification. Ces derniers causent la perte de la

dimension de la sécurité comme l'authentification, la non-répudiation, l'intégrité et la confidentialité. Par conséquent, la sécurité de l'information sera endommagée. Notre deuxième proposition et le protocole EIAKEP sont les plus sécurisés. Nous avons approuvé cette efficacité par l'analyse théorique et l'analyse formelle.

Références	Métriques de sécurité (nombre d'attaques)						
	M1	M2	M3	M4	M5	M6	M7
(Cheng, 2001; Zhou, 2000)	×	×	×	×	×	×	×
(Haddad et al., 2004)			×	×	×		
(Su and chang, 2007)			×	×	×		
(Zheng and Zhang, 2009)	×	×	×	×	×	×	
(Aiello et al., 2002)			×	×	×		×
(Ray et al., 2012)			×	×			
(Haddad and Mirmohamadi, 2005) IKEv2			×	×	×		
(La première proposition)	✓	✓	×	×	✓	✓	✓
(La deuxième proposition)*	✓	✓	✓	✓	✓	✓	✓
(EIAKEP)**	✓	✓	✓	✓	✓	✓	✓

Tableau 5.1 Étude comparative en terme de sécurité

Remarque: M1- Perfect Forward Security (PFS); M2- Known Key Security; M3- l'attaque Modification; M4- l'attaque par réflexion; M5- l'attaque par rejeu (Replay); M6- l'attaque DoS; M7- l'attaque de l'homme du milieu; ✓ signifie « satisfaite »et × « non satisfaite »; *- le nombre des messages utilisé sont 5; **- le nombre des messages utilisé sont 4.

En outre, la complexité des protocoles IKE existants dans la littérature ainsi que notre première proposition est élevée, car ils ont besoin de beaucoup de calculs modulaires pour créer la clé secrète afin de garantir un niveau de sécurité élevé. Cette synthèse nous a permis de proposer deux nouveaux protocoles qui vérifient plusieurs dimensions de sécurité (authentification, non-répudiation, l'intégrité et la

confidentialité) avec une clé de petite taille et de faible complexité ce qui rend nos protocoles très efficaces et sécurisés.

Dans le tableau 5.2, nous détaillons l'analyse de la complexité entre nos protocoles et les autres versions du protocole IKE.

Notre deuxième proposition du protocole IKE utilise: les opérations les plus rapides telles que (addition, soustraction et XOR); trois messages pour l'authentification mutuelle et la création SA-IKE; deux messages pour créer SA-IPSec. De plus, elle utilise une seule fois le cryptage symétrique avec une taille de clé plus courte. Donc, notre schéma peut fournir un niveau de sécurité élevé avec de meilleures performances. Dans le EIAKEP nous avons optimisé le nombre des échanges ce qui rend notre protocole plus efficace dans les réseaux de capteurs.

Références	P1	P2	P3	P4	P5	P6
(Cheng, 2001) IKE using pre-shared key I/R	*10/10*	*0/0*	*5/5*	*2/2 (exp) 0/0(sub)*	*0/0*	*0/0*
(Cheng, 2001) IKE using public key signature I/R	*10/10*	*0/0*	*6/6*	*2/2 (exp) 0/0 (sub)*	*0/0*	*1/1*
(Su and chang, 2007) I/R	*0/0*	*2/2*	*2/4*	*5/6 (exp) 0/0 (sub)*	*0/0*	*0/0*
(Haddad and Mirmohamadi, 2005) IKEv2 I/R	*6/6*	*0/0*	*3/3*	*2/2 (exp) 0/0 (sub)*	*0/0*	*1/1*
(Ray et al., 2012) I/R	*10/10*	*0/0*	*3/3*	*0/0(exp) 0/0(sub)*	*0/0*	*0/0*
La deuxième proposition I/R	*0/0*	*5/5*	*2/2*	*0/0 (exp) 1/1 (sub)*	*2/2*	*0/0*
EIAKEP I/R	*0/0*	*7/7*	*2/2*	*0/0 (exp) 1/1(sub)*	*0/0*	*0/0*

Tableau 5.2 L'analyse de la complexité entre nos protocoles et les autres versions du protocole IKE

Remarque: P1- la fonction Pseudo Radom; P2- fonction de hachage; P3- clé secrète dé/ chiffrement; P4- Calcul modulaire (exp, sub); P5- XoR; P6- clé publique dé/ chiffrement.

La figure 5.15 illustre une comparaison en termes du nombre de messages dans la phase I et la phase II entre nos protocoles proposés et trois autres protocoles existants dans la littérature.

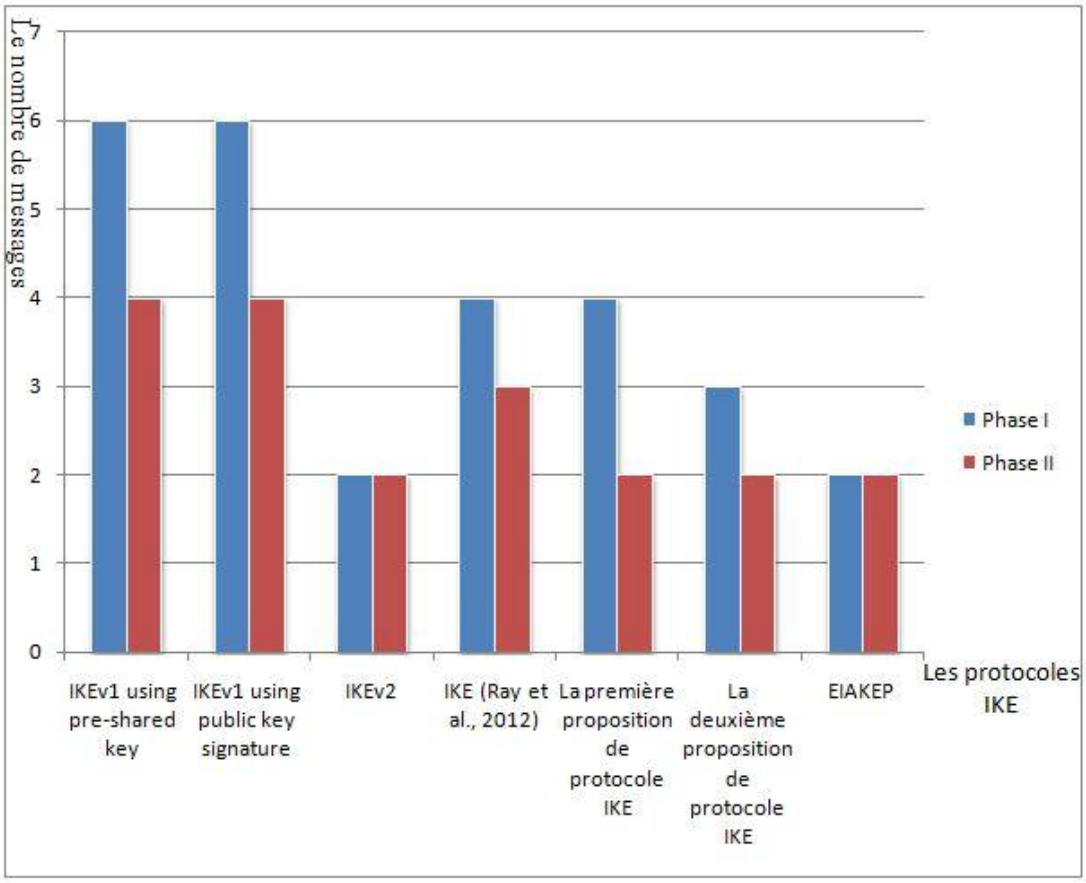

Figure 5.15 Étude comparative en terme de nombre de messages dans la phase I et la phase II

Dans les sections précédemment citées nous nous sommes basés sur la métrique de sécurité « le nombre d'attaques » pour évaluer la sécurité d'IPSec où nous avons concentré nos efforts sur la partie d'initialisation du protocole IPSec qui utilise le protocole IKE. Nous avons proposé des améliorations du protocole IKE pour renforcer la sécurité de l'IPSec. Dans la section suivante nous présentons une nouvelle approche de mesure de sécurité basée sur la politique de sécurité.

5.5. Une nouvelle approche de mesure de sécurité basée sur les politiques de sécurité

La sécurité de l'information et la protection de la vie privée sont devenue un problème crucial rencontré par les organisations. Différents protocoles ont été développés et appliqués pour protéger les données, les services et les ressources contre les attaques passives et actives, tels que le protocole IPSec afin d'assurer plusieurs dimensions de sécurité telle que la confidentialité, l'intégrité et l'authentification. Pour améliorer la sécurité des organisations, une approche de mesure de sécurité est nécessaire.

Il ya plus de 100 ans Lord Kelvin a souligné l'importance de la mesure. Il a dit, « *Si vous ne pouvez pas mesurer, vous ne pouvez pas améliorer* ». Donc, la mesure n'est pas un nouveau domaine de recherche, mais elle n'a pas attiré l'attention des chercheurs. Aujourd'hui, ce domaine est devenu l'un des axes les plus importants de la recherche scientifique. Les politiques de sécurité sont les outils les plus utilisés pour la mise en œuvre de la sécurité organisationnelle. Le tableau 5.3 présente une étude comparative entre les travaux connexes. La plupart des documents étudiés se concentrent sur l'analyse de la vulnérabilité et l'analyse de la menace, mais aucun travail dans la littérature ne fournit des métriques de sécurité qui permettent d'évaluer l'efficacité de la politique de sécurité.

Méthodologie / *Références*	Analyse de la menace	Analyse de la vulnérabilité		Attaque Propagation	Politique de sécurité
		existant	historique		
(Savola and Abie, 2009)	✓	×		×	×
(Ahmed et al.,2008)	×	✓	✓	✓	×
(Casola et al.,2007)	×	×	×	×	✓

Tableau 5.3 Étude comparative de travaux connexes

L'objectif de cette recherche est la conception d'un nouveau cadre de mesure de sécurité, qui repose sur la décomposition hiérarchique de la politique de sécurité. Nous avons proposé deux définitions une pour la métrique de sécurité et l'autre pour la politique de sécurité. Un ensemble de métriques de politique de sécurité est identifié et une évaluation quantitative de ces métriques est développée. L'approche proposée peut aider l'organisation à comparer ces politiques de sécurité avec d'autres

politiques, et de déterminer quelle politique de sécurité peut répondre à ces objectifs de sécurité.

5.5.1 Nos définitions proposées pour la métrique, la mesure et la politique de sécurité

- Les métriques de sécurité : sont des indicateurs mesurables des dimensions de la sécurité. elles sont utilisées pour fournir la prise de décision et d'améliorer l'efficacité de la sécurité de l'entité (système, produit, ou autre) à travers la collecte et l'analyse des données.
- La mesure de sécurité: est un processus de mesure qui fournit les informations sur l'efficacité de la sécurité.
- La politique de sécurité : est un ensemble de règles qui définissent les dimensions de sécurité souhaités pour chaque paquet en fonction de certains attributs tels que le protocole de sécurité, l'adresse source et l'adresse de destination.

5.5.2 L'approche proposée

Afin d'atteindre l'objectif de sécurité souhaitée par l'organisme. Un ensemble de mécanismes de sécurité sont intégrés. Ceux-ci sont configurés selon une politique de sécurité. Cependant, comme mentionné précédemment, une politique de sécurité est un ensemble de règles qui permettent d'atteindre une / ou plusieurs dimension(s) de sécurité. Dans notre étude, nous nous concentrons sur trois dimensions de sécurité: l'authentification, l'intégrité, la confidentialité. Par conséquent, les politiques de sécurité sont la base de la gestion de la sécurité d'une organisation.

Afin de vérifier et améliorer la sécurité de l'organisation, nous avons proposé une méthodologie d'évaluation de la sécurité. La figure 5.16 illustre l'emplacement du processus de mesure de sécurité dans un contexte organisationnel plus large. Le processus de mesure de sécurité comprend deux étapes:

1. L'identification et la définition: Tout d'abord, l'organisation identifie les dimensions de sécurité souhaitées par une analyse de la menace et de la vulnérabilité. Puis, un ensemble d'exigences de sécurité est indiqué d'après

l'analyse de la menace et de la vulnérabilité. Enfin, les politiques de sécurité sont décrits pour atteindre l'objectif de sécurité souhaité.

2. Le processus de développement de la métrique: celui-ci est basé sur la décomposition de la politique de sécurité. Cette dernière est configurée par l'administrateur, qui spécifie les dimensions de la sécurité de son réseau. Dans cette étude, nous sommes concentrés sur trois dimensions de la sécurité (la confidentialité, l'intégrité et l'authentification) qui représentent les objectifs de sécurité de plus de 100 organisations. Les étapes du développement des métriques de sécurité sont les suivantes:

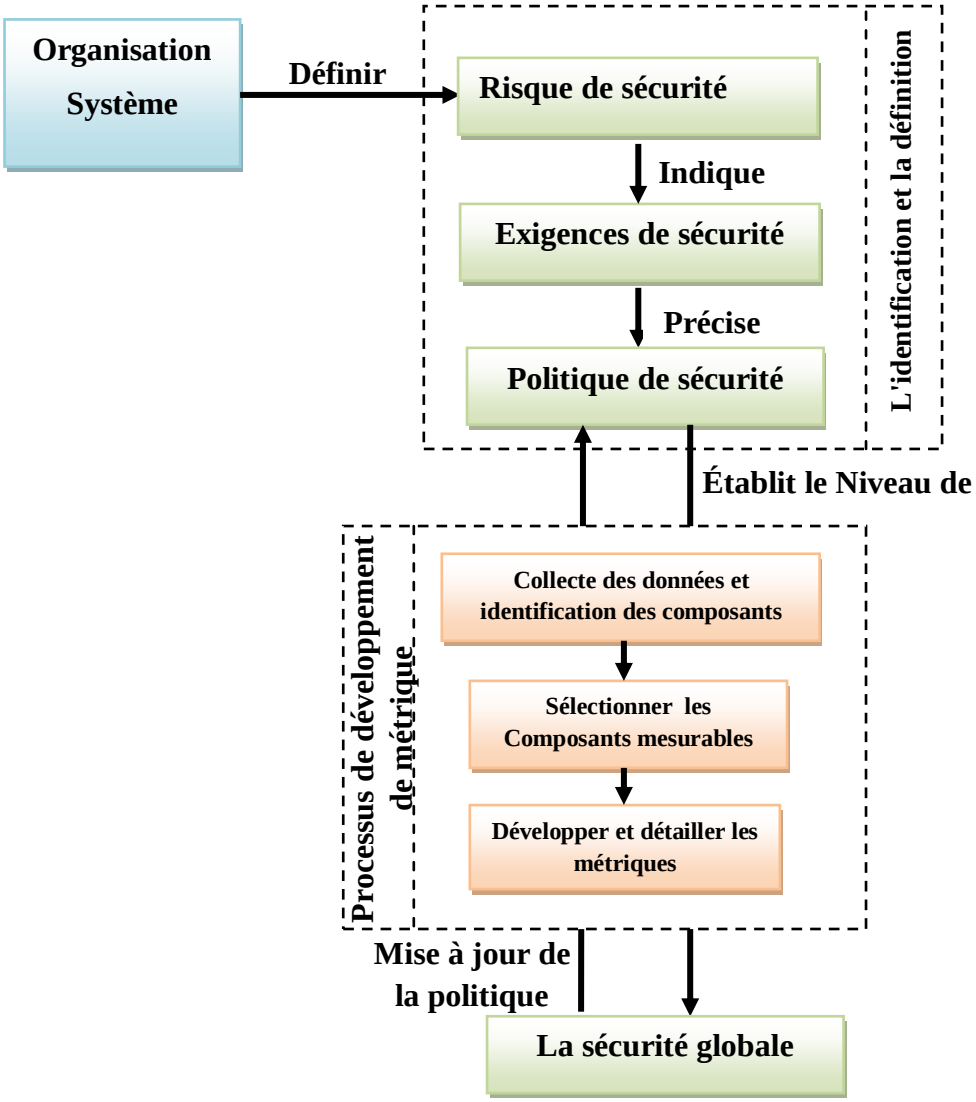

Figure 5.16 Le processus de mesure de sécurité proposé

- Identifier les propriétés de sécurité et classifier les politiques de sécurité en fonction de la dimension de la sécurité.

- Identifier les composants mesurables de chaque politique de sécurité en utilisant le processus de décomposition proposé dans (Wang et Wulf, 1997).

- Sélectionner les composants mesurables en se basant sur l'efficacité du mécanisme de sécurité, l'exactitude et l'efficacité de la structure de la politique de sécurité. Les composants mesurables sont les métriques de la politique de sécurité.

- Définir les détails de chaque métrique de sécurité et le niveau de sécurité global de l'organisation.

5.5.2.1 Décomposition de la politique de sécurité

Le cœur du processus de développement de la métrique de sécurité proposé est la décomposition de la politique de sécurité. La figure 5.17 illustre cette décomposition.

Comme le montre la figure 5.17, une classification en fonction des dimensions de sécurité de la politique (politique d'authentification, politique d'intégrité et politique de confidentialité) est effectuée. Pour trouver les éléments mesurables de chacune de ces politiques, un processus de décomposition est utilisé.

Ce dernier se termine lorsqu'aucun des nœuds de feuille ne peut être décomposé. Nous définissons le niveau de sécurité du système de sécurité de l'organisation par le niveau de la sécurité globale. Cette dernière dépend du niveau de sécurité locale de chaque politique de sécurité. Le niveau de sécurité locale est lié à la classification et à la décomposition de la politique. Dans ce qui suit, nous présentons les métriques de sécurité de chaque politique de sécurité ainsi que les mesures de ces métriques.

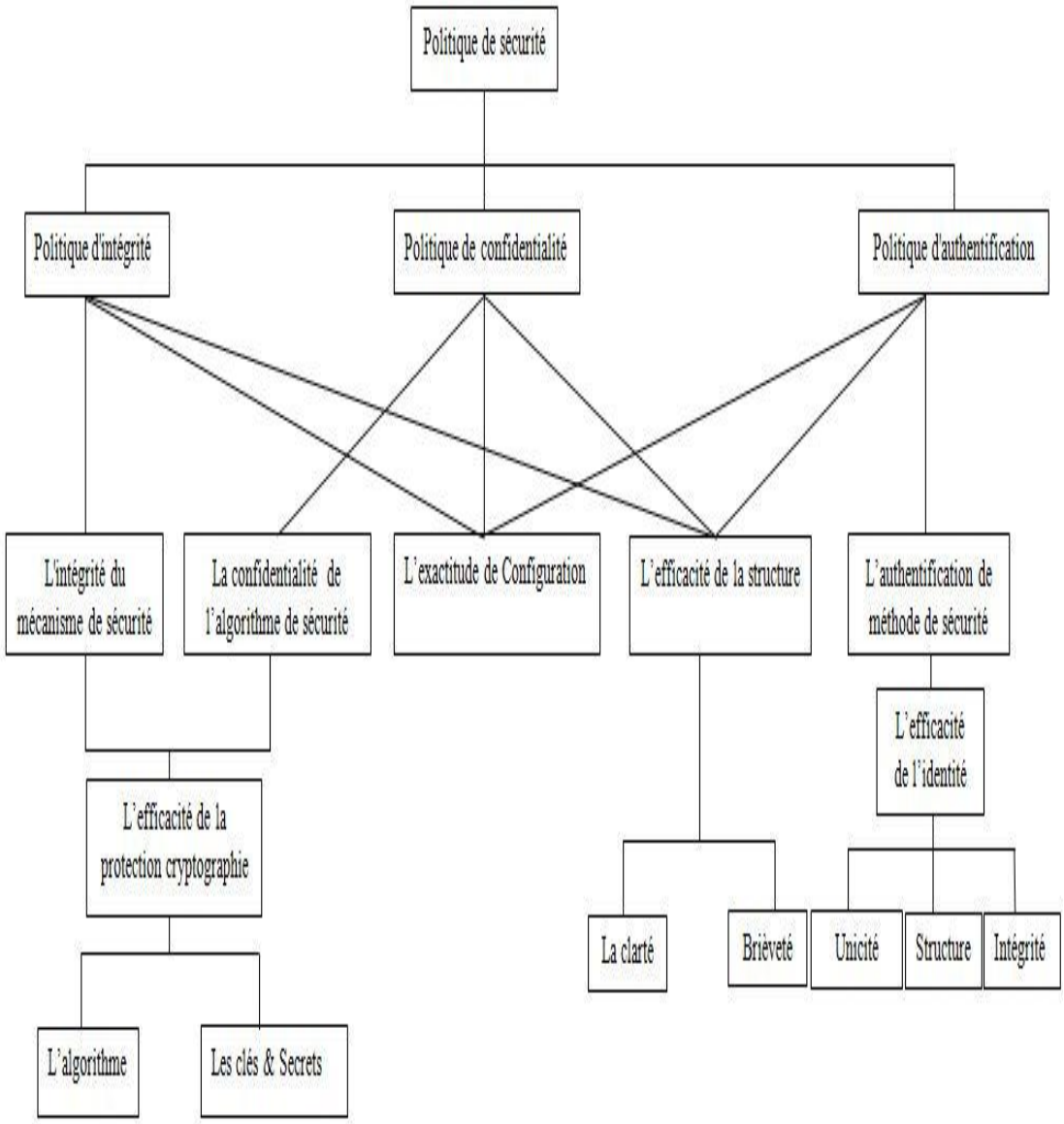

Figure 5.17 La décomposition de la politique de sécurité proposée

A. La politique de l'intégrité: elle sert à garantir que les informations transmises entre la source et la destination n'ont pas été modifiées. Le niveau de sécurité locale de la politique de l'intégrité (Local Security Level of the Integrity Policy, LSLIP) dépend de nombreux éléments mesurables tels que : l'intégrité du mécanisme de sécurité (Security Mechanism integrity, SMI), l'exactitude de configuration de la politique de l'intégrité (Configuration Correctness of Integrity Policy, CCIP) et l'efficacité de la structure de la politique de l'intégrité (Structure

Effectiveness of the Integrity Policy, SEIP). LSLIP est calculé par la somme pondérée des composants mesurables.

LSLIP= W_0. SMI+ W_1. CCIP + W_2. SEIP / W_i : est un facteur de pondération (1)

- L'intégrité du mécanisme de sécurité (SMI): dépend de la protection cryptographique de l'algorithme de l'intégrité (Cryptography Protection of the Integrity Algorithm, CPIA) et la protection cryptographique des clés et des informations confidentielles (Cryptography Protection of Keys & Secrets, CPKS). Les paramètres de sécurité de ceux-ci sont présentés dans le tableau 5.4 et 5.5 respectivement. Les équations 2, 3, 4 représentent respectivement la mesure de SMI, de CPIA et de CPKS.

$$SMI = W_0. CPIA + W_1. CPKS / W_i: \text{est un facteur de pondération} \quad (2)$$

$$CPIA = f(KL, AC, AN, AI) \quad (3)$$

$$CPKS = f(KL, MSMK) \quad (4)$$

Notation	***Métrique***
KL	Longueur de la clé (Key Length)
AC	La complexité de l'algorithme (Algorithm Complexity)
AN	Nombre d'attaques (Attacks Number)
AI	L'implémentation de l'algorithme (Algorithm Implementation)

Tableau 5.4 Les métriques de l'algorithme cryptographique (Jorstad and Landgrave, 1997)

Notation	***Métriques***
KL	Longueur de la clé (Key Length)
MSMK	La gestion et le mécanisme de stockage de clé (Management and Storage Mechanisms of Key)

Tableau 5.5 Les métriques de la protection des clés et des secrets

- L'exactitude de la configuration de la politique de l'intégrité (Configuration Correctness of Integrity Policy, SCCIP): la politique de sécurité configurée par un expert devra avoir à un risque faible par rapport à celle qui a été configurée par un débutant.

$$CCIP = Low, Medium, High \quad (5)$$

- L'efficacité de la structure de la politique de l'intégrité (Structure Effectiveness of the Integrity Policy, SEIP): dépend de la clarté de la structure de la politique de l'intégrité (CSIP) et la brièveté de la structure de la politique de l'intégrité (BSIP) (Goel and Chengalur-Smith, 2010).

  Les équations 6 et 7 représentent respectivement la mesure de SEIP et celle de BSIP.

  $$SEIP = W_0 . CSIP + W_1 \ BSIP / W_i$$: est un facteur de pondération (6)

  - La clarté de la structure de la politique de l'intégrité (CSIP): la mesure de la CSIP est obtenue par un questionnaire sur (Goel and Chengalur-Smith, 2010):
    1. La politique est facile à comprendre;
    2. La politique est facile à lire;
    3. La politique peut être comprise sans documents de référence.

  Il y a plusieurs façons de transformer un questionnaire en une métrique. L'une des méthodes possibles est d'utiliser seulement «oui» ou «non» pour répondre aux questions et attribuer un 1 à une réponse «oui» et de 0 à une réponse « non ». La mesure peut être dérivée en calculant le pourcentage de réponse «oui».

  - La brièveté de la structure politique de l'intégrité (BSIP): l'équation 7 présente la mesure de la BSIP telle que: le TN est le nombre total de mots; TNNU est le nombre de mots non unique; WTNNU est le facteur de pondération.

$$BSIP = \frac{TN}{TN + W_{TNNU} \ TNNU} \quad (7)$$

B. La politique de confidentialité: elle sert à assurer que l'information peut être utilisée ou lue par sa destination et non par une autre entité. Le niveau de sécurité locale de la politique de confidentialité (LSLCP) dépend de nombreux éléments mesurables tels que: La confidentialité du mécanisme de sécurité (Security Mechanism Confidentiality, SMC), l'exactitude de configuration de politique de la confidentialité (Configuration Correctness of Confidentiality Policy, CCCP) et l'efficacité de la structure de la politique de confidentialité (Structure

Effectiveness of the Confidentiality Policy, SECP). LSLCP est calculé par la somme pondérée des composants mesurables.

$$\textbf{LSLCP=} W_0.\ SMC + W_1.\ SCCCP + W_2.\ SECP /$$

W_i : est un facteur de pondération (8)

Les métriques de sécurité et la mesure de la SMC, CCCP et SECP sont semblables à SMI, CCIP et SEIP respectivement.

C. La politique d'authentification : elle sert à vérifier l'identité de l'entité qui demande l'accès aux ressources du système ou des applications. Le niveau de sécurité locale de la politique d'authentification LSLAP dépend de nombreux éléments mesurables tels que : la méthode de sécurité d'authentification (Security Method Authentication, SMA), l'exactitude de la configuration de la politique d'authentification (Configuration Correctness of the Authentication Policy, SCCAP) et l'efficacité de la structure de la politique d'authentification (Structure Effectiveness of the Authentication Policy, SEAP). LSLAP est calculé par la somme pondérée des composants mesurables (équation 9).

$$\textbf{LSLAP} = W_0\ SMA + W_1\ SCCAP + W_2\ SEAP /$$

W_i: est un facteur de pondération (9)

- La méthode de sécurité d'authentification (SMA): dépend de l'unicité de l'identité de la méthode d'authentification (UIMA), la structure identité de la méthode d'authentification (LMSI) et de l'intégrité de l'identité de la méthode d'authentification (IIMA). Les métriques de sécurité de ceux-ci ont été précédemment détaillées (chapitre 4). L'équation 10, présente la mesure de la SMA.

$$SMA = W_0.\ UIMA + W_1.SIMA + W_2.\ IIMA \quad (10)$$

Dans ce travail, nous avons proposé une nouvelle méthodologie de mesure de sécurité qui est basée sur la classification et la décomposition de la politique de sécurité. Nous avons classé la politique de sécurité selon les trois dimensions de la sécurité (confidentialité, authentification et intégrité). Ensuite, nous avons appliqué le processus de décomposition à chaque politique afin de trouver des mesures de

sécurité. Nous avons identifié plusieurs techniques pour mesurer les métriques de sécurité dans le but d'évaluer le niveau d'un système de sécurité.

5.6. Conclusion

Dans ce chapitre, nous avons proposé trois améliorations pour le protocole IKE afin de renforcer la sécurité de l'IPSec. Nous avons utilisé la métrique de sécurité « le nombre d'attaques » pour évaluer la sécurité d'IPSec.

Dans la première proposition du protocole IKE, nous avons focalisé nos efforts sur trois types d'attaque: DoS, l'homme du milieu et par rejeu. Notre protocole peut résister à ces attaques, mais il ne satisfait pas plusieurs exigences de sécurité; en plus, la complexité du protocole est élevée. Cette synthèse, nous a permis de proposer un nouveau protocole IKE basé sur la cryptographie à courbe elliptique, ce dernier satisfait plusieurs exigences de sécurité et résiste à plusieurs types d'attaques telles que: DoS, l'homme du milieu, par rejeu, par réflexion. De plus, il assure un niveau de sécurité élevé avec une clé de petite taille et avec moins de calcul modulaire. Il utilise trois messages pour créer l'association de sécurité IKE et deux messages pour établir l'association de sécurité IPSec. Afin de rendre le protocole IKE plus léger et très efficace dans les réseaux de capteurs, nous avons proposé un autre schéma pour le protocole IKE qui s'appelle EIAKEP. L'EIAKEP assure un niveau de sécurité très élevé et il n'utilise que deux messages pour établir le SA-IKE et deux messages pour le SA-IPSec.

D'autre part, nous avons proposé une nouvelle méthodologie de mesure de sécurité pour l'IPSec basée sur la classification et la décomposition de la politique de sécurité. Nous avons classé la politique de sécurité selon les trois dimensions de sécurité (confidentialité, authentification et intégrité). Ensuite, nous avons appliqué le processus de décomposition à chaque politique afin de trouver les métriques de sécurité. Nous avons identifié plusieurs techniques pour mesurer les métriques de sécurité dans le but d'évaluer le niveau de sécurité du système. Dans l'attente d'une confirmation formelle nous jugeons que notre proposition peut s'appliquer non seulement à l'IPSec, mais aussi à tous les protocoles qui se basent sur les politiques de sécurité.

Conclusion générale

Parmi les différents protocoles de sécurité réseau, nous trouvons le protocole IPSec. Ce protocole est devenu indispensable dans le réseau internet. Il nous permet de fournir une communication sécurisée entre les entités communicantes. Il assure plusieurs dimensions de sécurité telles que: l'authentification de la source de données, la confidentialité, l'intégrité des données et le contrôle d'accès.

La première version de l'IPSec était basique, elle a été développée sous forme de RFC (Request For Comment) sans la partie de gestion de clé. Cette version a montré beaucoup de problèmes et défauts. Afin de pallier à ces problèmes et défauts, une première amélioration de ce protocole a été développée en 1998, où ils ont proposé le protocole IKEv1, qui permet d'ajouter un système dynamique pour la gestion des paramètres confidentiels de l'IPSec. Malheureusement, cette première version a été critiquée par plusieurs chercheurs en particulier la vulnérabilité aux attaques passives et actives.

1. Les contributions

Dans notre thèse, nous avons proposé trois améliorations du protocole IKE afin de renforcer la sécurité de l'IPSec. Nous avons utilisé la métrique de sécurité « le nombre d'attaques » pour évaluer la sécurité de l'IPSec.

La première proposition du protocole IKE a été basée sur trois types d'attaques : DoS, l'homme du milieu et par rejeu. Notre protocole peut résister à ces attaques, mais il ne satisfait pas plusieurs exigences de sécurité, en plus la complexité du protocole est élevée. Cette synthèse, nous a permis de proposer un nouveau protocole IKE basé sur la cryptographie à courbe elliptique, ce dernier satisfait plus d'exigences de sécurité et résiste à plusieurs types d'attaques telles que : DoS, l'homme du milieu, par rejeu, par réflexion..etc. De plus, il assure un niveau de sécurité plus élevé avec une clé de petite taille et moins de calcul modulaire. Il utilise trois messages pour créer l'association de sécurité IKE et deux messages pour établir l'association de sécurité de l'IPSec. Afin de rendre ce protocole IKE plus léger et très efficace dans les réseaux de capteurs. Nous avons proposé un autre schéma du protocole IKE nommé EIAKEP. L'EIAKEP assure un niveau de sécurité très élevé et

il n'utilise que deux messages pour établir le SA-IKE et deux messages pour établir le SA-IPSec.

D'autre part, nous avons proposé une nouvelle méthodologie de mesure de sécurité pour l'IPSec qui se base sur la classification et la décomposition des politiques de sécurité. Nous avons classé les politiques de sécurité selon trois dimensions de sécurité (confidentialité, l'authentification et l'intégrité). Ensuite, nous avons appliqué le processus de décomposition à chaque politique afin de trouver les métriques de sécurité. Nous avons identifié plusieurs techniques pour mesurer les métriques de sécurité dont le but est d'évaluer le niveau de sécurité du système. Dans l'attente d'une confirmation formelle nous jugeons que notre proposition peut s'appliquer non seulement à l'IPSec, mais aussi à tous les protocoles qui se basent sur les politiques de sécurité.

2. Perspectives

Dans nos futurs travaux, nous allons tester l'efficacité de nos contributions dans les réseaux de capteurs. Nous envisageons aussi l'adaptation de notre protocole EIAKEP dans d'autres protocoles de sécurité. De plus, nous projetons l'implémentation de notre approche de mesure de sécurité sur les réseaux qui utilisent les politiques de sécurité à l'image de TLS.

Références

(Ahmed et al, 2008) Ahmed, M.S., Al-Shaer, E. and Khan, L. (2008) "A Novel Quantitative Approach For Measuring Network Security", In Proceedings of 27th IEEE Communications Society Conference on Computer Communications, Phoenix, AZ, pp.76-80.

(Aiello et al., 2002) Aiello, W., Bellovin, S.M., Blaze, M., Canetti, R., Ioannidis, J., Keromytis, A.D. and Reingold, O. (2002) "Efficient, DoS Resistant, Secure Key Exchange for Internet Protocols", In Proceedings of the 9th ACM conference on Computer and communications security, Washington, USA, pp.48-58.

(Allard et al., 2008) Allard, F. and Bonnin, J.M. (2008) "An application of the context transfer protocol: IPsec in a IPv6 mobility environment", Communication Networks and Distributed Systems, Vol.1, No.1, pp.110-126.

(Anderson et al.,) Anderson, R., Stajano, F. and Lee, J.H., "Security Policies", [online] http://www.cl.cam.ac.uk/~rja14/Papers/security-policies.pdf [consulter en janvier 2013].

(Atzeni and Lioy, 2005) Atzeni, A. and Lioy, A. (2005) "Why to adopt a security metric? a little survey". In First Workshop on Quality of Protection Quality of Protection workshop, Milan, Italy, pp.1-12.

(AVISPA, 2006) AVISPAv1.1. (2006) "User Manual", [online] http://www.avispa-project.org/ [consulter en janvier 2013].

(Barman, 2001) Barman, S. (2001), "Writing Information Security Policies", New Riders Publishing Thousand Oaks, CA, USA, ISBN: 157870264X.

(Bartol et al., 2009) Bartol, N., Bates, B., Goertzel, K.M. and Winograd, T. (2009) "Measuring cyber security and information assurance: a state-of-the-art report," Information Assurance Technology Analysis Center IATAC, [online] https://buildsecurityin.uscert.gov/sites/default/files/MeasuringCybersecurityIA.PDF [consulter en janvier 2014].

(Basu et al., 2012) Basu, A., Sengupta, I. and Kanta-Sing, J. (2012) "Formal Security Verification of Secured ECC Based Signcryption Scheme", In Proceedings of the Second International Conference on Computer Science, Engineering & Applications, New Delhi, India, pp.713-725.

(Baudet, 2007) Baudet, M. (2007) "Sécurité des protocoles cryptographiques: aspects logiques et calculatoires", l'École Normale Supérieure de Cachan, [online] https://tel.archives-ouvertes.fr/tel-00140916 [consulter en janvier 2012].

(Blake, 2005) Blake, F. (2005) "Advances in Elliptic Curve Cryptography", United States of America by Cambridge University Press, New York, ISBN-13: 978-0-521-60415-4.

(Bouamama et al., 2008) Bouamama, M.N and Xerin, A. (2008) "Sécurité dans TCP/IP", Université de Reims Champagne-Ardenne, [online] http://docplayer.fr/555721-Universite-de-reims-champagne-ardenne-https-ssl-ssh-ipsec-et-socks-presente-par-bouamama-mohamed-nadjib-aziz-xerin.html [consulter en janvier 2014].

(Caballero, 2008) Caballero, X.F. (2008) "Etude d'IPSec: projet d'une API-IPSec pour la mobilité et le multihoming", Télécom Recherche et Développement, France,[online]http://upcommons.upc.edu/bitstream/handle/2099.1/6862/PFC%20Xavier_Ferrer.pdf?sequence=1 [consulter en janvier 2012].

(Casola et al., 2007) Casola, V., Mazzeo, A., Mazzocca, N. and Valeria Vittorini (2007) "A policy-based methodology for security evaluation: A Security Metric for Public Key Infrastructures", Computer Security, Vol. 15, No. 2, pp.197-229.

(Cheng et al., 2001)Cheng, P.C, Garay, J. A., Herzberg, A. and Krawczyk, H. (2001) "An architecture for internet key exchange protocol", IBM System, Vol. 40, No. 3, pp.721-746.

(Cole et al., 2005) Cole, E., Krutz, R., Conley, J. (2005), "Network Security Bible", Wiley Publishing, Inc, ISBN-13:978-0-7645-7397-2.

(Cremers, 2011) Cremers, C. (2011) "Key Exchange in IPsec revisited: Formal Analysis of IKEv1 and IKEv2", In Proceedings of 16th European Symposium on Research in Computer Security, Leuven, Belgium, pp.315-334.

(Debbeche et al., 2008) Debbeche, F., Ghoualmi-Zine, N. (2008) " Système Acoustico-Anatomique pour l'Identification des Locuteurs par Localisation dans un Espace de Locuteurs de Référence ", Journée Jeunes Chercheurs en Informatique, Guelma, pp.21-24.

(Denizot et al., 2015) Denizot, E., Pereira, J. and Berger, A. "VPN: Virtual Private Network niveau 2 et niveau 3", [online] http://docplayer.fr/1618383-Eric-denizot-jose-pereira-anthony-berger.html [consulter en janvier 2015].

(Doraswamy and Harkin, 2003) Doraswamy, N., Harkin, D. (2003) "IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks (2nd Edition)", Prentice Hall, États-Unis. ISBN: 007-6092018759.

(El-Gamal, 1985) El-Gamal, T. (1985) "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol.31, No.4, pp.469-472.

(Firesmith, 2004) Firesmith, D., (2004) "Specifying Reusable Security Requirements", Object Technology, Vol.3, No.1, pp.61-75.

(Frankel, 2001) Frankel, S. (2001) "Demystifying the IPsec Puzzle", Artech House Publishers, London, ISBN: 978-1580530798.

(Glouche et al., 2006) Glouche, Y., Genet, T., Heen, O. and Courtay, O. (2006) "A Security Protocol Animator Tool for AVISPA", In Workshop on Security Specification and Verification of Embedded Systems, Pisa, pp.1-7.

(Goel and Chengalur-Smith, 2010) Goel, S., and Chengalur-Smith I. N. (2010) "Metrics for characterizing the form of security policies", Strategic Information Systems , Vol. 19, No. 4, pp. 281-295.

(Guillot, 2013) Guillot, P. (2013) "La cryptologie : L'art des codes secrets ", EDP sciences, France, ISBN : 978-2-7598-0811-3.

(Haddad and Mirmohamadi, 2005) Haddad, H. and Mirmohamadi, H. (2005) "Comparative evaluation of successor protocols to Internet key exchange IKE", In Proceedings of the IEEE Intl. Conf. on Industrial Informatics, Perth, Australia, pp.692-696.

(Haddad et al., 2004) Haddad, H., Berenjkoub, M. and Gazor, S. (2004) "A proposed protocol for Internet Key Exchange (IKE)", In Proceedings of Electrical and Computer Engineering, Niagara Falls, Canada, pp.2017-2020.

(Hajjeh, 2004) Hajjeh, I. (2004) "Conception et validation d'un nouveau protocole pour la sécurisation des échanges", Ecole Nationale Supérieure des Télécommunications, Paris, [online] http://igm.univ-mlv.fr/~duris/NTREZO/20042005/Lamotte-Robert-Seigneurin_SSH-TLS.pdf [consulter en janvier 2012].

(Hallberg et al., 2011) Hallberg, J., Eriksson, M., Granlund, H., Kowalski, S., Lundholm, K., Monfelt, Y., Pilemalm, S., Wätterstam, T. and Yngström, L.

(2011) "Controlled Information Security", Stockholm University, ISSN:1650-1942.

(Harkins and carrel, 1998) Harkins, D. and carrel, D. (1998) "RFC2409: The Internet Key Exchange (IKE)", IETF, [online] http://www.ietf.org/rfc/rfc2409.txt [consulter en janvier 2012].

(Hayden, 2010) Hayden, L. (2010) "IT Security Metrics: A Practical Framework for Measuring Security & Protecting Data", The McGraw-Hill Companies, ISBN-13: 978-0071713405.

(ISO/IEC 27000, 2009) ISO/IEC 27000 (2009) "Information technology — Security techniques — Information security management systems — Overview and vocabulary ", [online] http://standards.iso.org/ittf/licence.html [consulter en janvier 2012].

(Iso-Anttila et al., 2007) Iso-Anttila, L., Ylinen, J. and Loula, P. (2007) "A proposal to improve IKEv2 negotiation", In Proceedings of the International Conference on Emerging Security Information, Systems, and Technologies, Valencia, pp.169-174.

(ISS, 2008) ISS: Innovation Security Solutions (2008) "Le protocole SSH (Secure Shell)", [online] http://ineovation.fr/white-papers/Secure_Shell.18042008.pdf [consulter en janvier 2014].

(ITU-T, 2003) ITU-T Recommendation X.805 (2003) "Security architecture for systems providing end-to-end communications", [online] https://www.itu.int/rec/T-REC-X.805-200310-I/en [consulter en janvier 2012].

(Jansen, 2009) Jansen.,W (2009) "Directions in Security Metrics Research" NIST, NISTIR 7564, pp.1-21.

(Johnson et al, 2011) Johnson, A., Dempsey, K., Ross, R., Gupta, S., Bailey, D. (2011), "Guide for Security-Focused Configuration Management of Information Systems", NIST, NIST Special Publication 800-128, pp.1-88.

(Jorstad and Landgrave, 1997) Jorstad, N. and Landgrave,T. S. (1997) "Cryptographic algorithm metrics", In Proceedings of 20th National Information Systems Security Conference, Baltimore, pp.1-38.

(Kaufman et al., 2010) Kaufman, C., Homan, P., Nir, Y. and Eronen, P. (2010) "RFC 5996: Internet Key Exchange Protocol Version 2 (IKEv2)", IETF, [online] http://www.rfc-editor.org/info/rfc5996 [consulter en janvier 2012].

(Kaufman, 2005) Kaufman, C. (2005) "RFC 4306: Internet Key Exchange (IKEV2) Protocol", IETF, [online] https://tools.ietf.org/html/rfc4306 [consulter en janvier 2012].

(Kent and Atkinson, 1998a) Kent, S. and Atkinson, R. (1998a) "RFC 2402: IP Authentication Header (AH)", IETF, [online] http://www.ietf.org/rfc/rfc2402.txt [consulter en janvier 2012].

(Kent and Atkinson, 1998b) Kent, S. and Atkinson, R. (1998b) "RFC 2406: IP Encapsulating Security Payload (ESP)", IETF [online] http://www.ietf.org/rfc/rfc2406.txt [consulter en janvier 2013].

(Kent and Atkinson, 1998c) Kent, S. and Atkinson, R. (1998c) "RFC 2401: Security Architecture for the Internet Protocol", IETF, [online] http://www.ietf.org/rfc/rfc2401.txt [consulter en janvier 2012].

(Kozierok, 2005) Kozierok, M.C. (2005) "The TCP/IP Guide", [online] http://www.tcpipguide.com/free/t_IPSecAuthenticationHeaderAH.htm [consulter en janvier 2012].

(Krautsevich et al., 2010) Krautsevich, L., Martinell, F. and Yautsiukhin, A. (2010) "Formal approach to security metrics:What does "more secure" mean for you?", In Proceedings of the Fourth European Conference on Software, New York, USA, pp. 162-169.

(Labouret, 2000) Labouret, G. (2000) "IPSec: présentation technique",Hervé schauer consultants, France, [online] http://www.hsc.fr/ [consulter en janvier 2012].

(Lafourcade, 2006) Lafourcade, P. (2006) "Vérification de protocoles cryptographiques en présence de théories équationnelles" l'École Normale Supérieure de Cachan, [online] https://tel.archives-ouvertes.fr/tel-00133494/ [consulter en janvier 2012].

(Lamotte et al., 2005) Lamotte, B., Robert, V., and Seigneurin, A. (2005) "Nouvelles technologies réseaux SSH et TLS", France, [online] http://igm.univ-mlv.fr/~duris/NTREZO/20042005/Lamotte-Robert-Seigneurin_SSH-TLS.pdf [consulter en janvier 2013].

(Lasserre and Klein, 2011) Lasserre, X. and Klein, T. (2011) "Réseaux Privés Virtuels–Vpn", [online] http://www.frameip.com/vpn/ [consulter en janvier 2012].

(Leon and Saxena, 2010) Leon, P.G. and Saxena, A. (2010) "An approach to quantitatively measure Information security", In Proceedings of 3rd India Software Engineering Conference, Mysore.

(Li, 2003) Li, M. (2003) "Policy-Based IPsec Management", IEEE Network, Vol.17, No.6, pp.36-43.

(Li, 2010) Li, N. (2010) "Research on Diffie-Hellman Key Exchange Protocol", In Proceedings of 2nd Computer Engineering and Technology (ICCET), Chengdu, China, pp.634 -637.

(Lu et al., 2008) Lu, N., Zhou, H. and Qin, Y. (2008) "A Comparison Study of IKE Protocols", In Proceedings of the International Conference on Mobile Technology, Applications, and Systems, New York, USA, pp.88-92.

(Mansour, 2013) Mansour, I. (2013) "Contribution à la sécurité des communications des réseaux de capteurs sans fil", université BLAISE PASCAL–CLERMONT II, [online] https://tel.archives-ouvertes.fr/tel-00877033/document [consulter en janvier 2014].

(Martin, 2006) Martin, B. (2006) " IPSec:Techniques " [online] http://www.madpowah.org/textes/38_52_ipSEC_FR.pdf [consulter en janvier 2012].

(Michael and Herbert, 2009) Michael, E.W. and Herbert J. M. (2009) "Principles of Information Security", Course Technology Cengage Learning, USA, ISBN-13: 9781111138219.

(Microsoft, 2005) Groupe Microsoft (2005) "IPSec Protocol Types", [online] https://technet.microsoft.com/en-us/library/cc757712%28v=ws.10%29.aspx [consulter en janvier 2012].

(Moore et al., 2001) Moore, B., Ellesson, E., Strassner, J., Westerinen, A. (2001) "RFC 3060: A. Policy Core Information Model", [online] https://tools.ietf.org/html/rfc3060 [consulter en janvier 2013].

(Nagalakshmi et al., 2011) Nagalakshmi, V., Rameshbabu, I. and Avadhani, P.S. (2011) "Modified protocols for internet key exchange (IKE) using public encryption key and signature keys", In Proceedings of the eighth international conference on Information Technology: New Generations, Las Vegas, NV, pp.376-381.

(NIST, 2013) NIST, (2013) http://www.chipestimate.com/tech _talks/2008/12/02/ Certicom-ECC-Drives-Next-Generation-Hardware-Security-Applications [Consulter en juin 2013].

(Nitaj, 2011) Nitaj, A. (2011) "La cryptographie du futur", Université de Caen, France [online] http://www.math.unicaen.fr/~nitaj [consulter en janvier 2012].

(Oppliger, 2009) Oppliger, R. (2009) "SSL and TLS: Theory and Practice", Artech House, Suisse, ISBN: 9781596934481.

(Perlma and Kaufman, 2001) Perlma, R. and Kaufman, C. (2001) "Analysis of the IPSec Key Exchange Standard", In Proceedings of Tenth IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, Cambridge, USA, pp.150-156.

(Peyrin, 2008) Peyrin, T. (2008) "Analyse de fonctions de hachage cryptographiques", Université de Versailles Saint-Quentin-en-Yvelines, [online] https://tel.archives-ouvertes.fr/tel-00767028/document [consulter en janvier 2012].

(Ray et al., 2012) Ray, S., Nandan, R. and Biswas, G.P. (2012) "ECC Based IKE Protocol Design for Internet Applications", In Proceedings of 2nd International Conference on Computer, Communication, Control and Information Technology of Technology (Elsevier), pp.522-529.

(Sahinoglu, 2005) Sahinoglu, M. (2005) "Security meter: A practical decision-tree model to quantify risk", IEEE Security and Privacy, Vol.3, No.3, pp.18-24.

(Savola and Abie, 2009) Savola, R. and Abie, H. (2009) "Identification of Basic Measurable Security Components for a Distributed Messaging System", In Proceedings of Third International Conference on Emerging Security Information, Systems and Technologies, Athens, Glyfada, pp. 121-128.

(Schneider, 2000) Schneider, F.B., (2000) "Enforceable Security Policies", ACM Transactions on Information and System Security, Vol.3, No.1, pp. 30-50.

(Shirley, 2006) Shirley C. P. (2006) "A Guide to Security Metrics, SANS Security Essentials GSEC Practical Assignment", Version 1.2e, [online] http://www.sans.org/reading_room/whitepapers/auditing/55.php [consulter en janvier 2013].

(Solms and Solms, 2004) Solms, R.V. and Solms, B.V (2004) "From policies to culture", Computers & Security, Vol.23, No. 4, pp.275-279.

(SSE-CMM, 2008) SSE-CMM: Systems Security Engineering Capability Maturity Model, International Systems Security Engineering Association (ISSEA) (2008), [online] http://www.sse-cmm.org/metric/metric.asp[consulter en janvier 2013].

(Stallings, 2014) Stallings, W. (2014) "Protocol Basics: Secure Shell Protocol", Internet Protocol, Vol.12, No.4.

(Stephen, 2000) Stephen, A. (2000) "SSL & TLS Essentials Securing the Web" John Wiley & Sons, Inc, New York, ISBN: 9780471383543.

(Stoddard et al., 2005) Stoddard, M., Bodeau, D., Carlson, R., Glantz, C. , Haimes, Y. , Lian, C. , Santos, J. and Shaw, J. (2005) "Process Control System Security Metrics – State of Practice", I3P–Institute for information infrastructure protection, No.1.

(Su and Chang, 2007) Su, M. and Chang, J.F. (2007) "An efficient and secured internet key exchange protocol design", In Proceedings of the fifth annual conference on Communication Networks and Services Research (CNSR'07), Fredericton, New Brunswick, Canada, pp.184-192.

(Swanson et al., 2003) Swanson, M., Bartol, N., Sabato, J., Hash, J. and Graffo, L. (2003) "Security Metrics Guide for Information Technology Systems", NIST Special Publication 800-55.

The American Heritage Dictionary. 4th ed. Houghton Mifflin Publishing Company (2000).

(Thomas and Elbirt, 2006) Thomas, J. and Elbirt, A.J. (2006) "Understanding Internet Protocol Security", Information Systems Security, Vol. 13, No. 4, pp.39-43.

(Thuillet, 2012) Thuillet, C. (2012) " Implantations cryptographiques sécurisées et outils d'aide à la Validation des contre mesures contre les attaques par canaux cachés ", l'université Bordeaux I, [online] http://www.theses.fr/2012BOR14508 [consulter en janvier 2012].

(Van Quang, 2005) Van Quang, D. (2005) "Contribution à l'étude de la qualité de service pour les protocoles sécurisés de télécommunications. Application à IPSec", l'université Paris XII-Val Marne, [online] http://doxa.u-pec.fr/theses/th0231084.pdf [consulter en janvier 2012].

(Venkata Krishna et al., 2013) Venkata Krishna, P., Misra, S., Joshi, D., Gupta, A. and Obaidat, M.S. (2013) "Secure socket layer certificate verification: a learning

automata approach", Security and Communication Networks, [online] http://onlinelibrary.wiley.com/doi/10.1002/sec.867/abstract [consulter en janvier 2014].

(Venter and Eloff, 2003) Venter, H.S. and Eloff, J.H.P. (2003) "A taxonomyfor information security technologies", Computers & Security, Vol. 22, No. 4, pp.299-307.

(Wang and Wulf, 1997) Wang, C. and Wulf, W.A. (1997) "Towards a Framework for Security Measurement", In Proceedings of 20th National Information Systems Security Conference, Baltimore, MD, USA, pp.522-533.

(Westerinen et al., 2001)Westerinen, A., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J. and Waldbusser, S. (2001) "RFC 3198: Terminology for Policy-Based Management", [online] https://www.ietf.org/rfc/rfc3198.txt [consulter en janvier 2013].

(Xenos, 2006) Xenos, M. (2006) "Software Metrics and Measurements", Encyclopedia of E-Commerce, E-Government and Mobile Commerce", Idea Group Publishing, ISBN: 1-59140-799-0, pp. 1029-1036.

(Ylonen et al., 2006a) Ylonen, T. and Lonvick, C. (2006) "RFC 4251: The Secure Shell (SSH) Protocol Architecture", [online] http://www.snailbook.com/docs/architecture.txt [consulter en janvier 2014].

(Ylonen et al., 2006b) Ylonen, T. and Lonvick, C. (2006) "RFC 4253: The Secure Shell Transport Layer Protocol", [online] https://www.ietf.org/rfc/rfc4253.txt [consulter en janvier 2014].

(Ylonen et al., 2006c)Ylonen, T. and Lonvick, C. (2006) "RFC 4252: The Secure Shell Authentication Protocol", [online] https://www.ietf.org/rfc/rfc4252.txt [consulter en janvier 2014].

(Ylonen et al., 2006d) Ylonen, T. and Lonvick, C. (2006) "RFC 4254: The Secure Shell (SSH) Connection Protocol", [online] https://www.ietf.org/rfc/rfc4254.txt [consulter en janvier 2014].

(Zeyad et al., 2011) Zeyad, M., Chien-Lung, H., Yaw-Chung, C. and Chi-Chun, L. (2011) "An efficient and secure three-pass authenticate key agreement elliptic curve based protocol", Innovative Computing, Information and Control, Vol. 7, No. 3, pp.1273-1284.

(Zheng and Zhang, 2009) Zheng, L. and Zhang, Y. (2009) "An Enhanced IPSec Security Strategy", In Proceedings of International Forum on Information Technology and Applications, China, pp.499-502.

(Zhou, 2000) Zhou, J. (2000) "Further analysis of the Internet key exchange protocol", Computer Communications, Vol. 23, No. 17, pp.1606-1612.

(Zhu et al., 2010) Zhu, x., Haigang, Z. and Jun, L. (2010) "Analysis and Improvement of IKEv2 against Denial of Service Attack", In Proceedings of International Conference on Information, Networking and Automation (ICINA), Kunming, pp.350-355.

Annexe : Liste des publications

(Ahmim et al., 2013) Ahmim, M., Babes,M. and Ghoualmi, N. "Contribution to enhance IPSec security by a safe and efficient internet key exchange protocol", ICMASM'2013, 2013 IEEE ISBN (Print) 978-1-4799-0460-0, ISBN (online) 978-1-4799-0462-4.

(Ahmim et al., 2014) Ahmim, M., Babes,M. and Ghoualmi, N. "Contribution in Authentication Systems Evaluated by Metrics of Security", Colloque sur l'Optimisation et les Systèmes d'Information COSI'2014, Béjaia, Algérie.

(Ahmim et al., 2013) Ahmim, M., Babes,M. and Ghoualmi, N. "Contribution to enhance IPSec security by a safe and efficient internet key exchange protocol", International Journal of Information Security Research, Volume: 4, Issue: 3, Page: 123-132.

(Ahmim et al., 2015) Ahmim, M., Babes,M. and Ghoualmi, N. "Formal analysis of efficiency and safety in IPSec based on internet key exchange protocol", Int. J. Communication Networks and Distributed Systems, Vol. 14, No. 2, 2015.

(Ahmim et al., 2016) Ahmim, M., Babes,M. and Ghoualmi, N. "Contribution on security metrics based on security policy" , Accepté par la conférence "International Conference on Software Engineering and Technologies" , Barcelona, Spain March 18 - 20, 2016.

(Ahmim et al., 2016) Ahmim, M., Babes,M. and Ghoualmi, N."EIAKEP: An efficient internet authenticated-key exchange protocol" [En cours d'exam].