Architecture de Certification Distribuée à base de Multi-signature

Mawloud Omar, Yacine Challal, Abdelmadjid Bouabdallah

Architecture de Certification Distribuée à base de Multi-signature

Mawloud Omar (mawloud.omar@gmail.com)*
Yacine Challal (yacine.challal@hds.utc.fr)†
Abdelmadjid Bouabdallah (bouabdal@hds.utc.fr)†

Abstract:
Dans cet article, nous proposons une architecture d'un modèle de confiance à base de certification pour les réseaux ad hoc mobiles. Le modèle permet la coexistence de plusieurs autorités de certification hétérogènes. Le rôle de chaque autorité de certification est assuré par un ensemble de serveurs qui signent collectivement les certificats aux utilisateurs. La signature des certificats est basé sur le mechanisme de chiffrement à base d'identité. Chaque autorité de certification supervise une communauté d'utilisateurs, dénommée *domaine*, où elle exécute ses propres règles de certification. Le passage d'un domaine à un autre se fait à l'aide d'un graphe de confiance établi entre les différentes autorités de certification. Pour évaluer les performances de notre modèle, nous avons effectué des simulations à travers les quelles nous avons étudié l'impact du nombre d'autorités de certification et le nombre de serveurs impliqués dans chacune d'elles.

Keywords: Sécurité, Confiance, Certificat, Multisignature, Réseau ad hoc mobile

1 Introduction

Pour assurer les services de sécurité dans un réseau ad hoc mobile, on doit s'appuyer sur un *modèle de confiance*. Ce dernier fournit un cadre de travail pour la construction et l'administration de la relation de confiance entre les nœuds dans un réseau. Selon l'ITU-T, le terme «confiance» est défini comme suit : «*On dit qu'une entité fait confiance à une deuxième entité si et seulement si cette dernière se comporte exactement comme la première le prévoit*» [15]. La gestion de la confiance dans les réseaux ad hoc fait l'objet de deux grandes catégorie de modèles : *les modèles de confiance à base de coopération* [16, 17, 18, 19, 20, 21, 22], et *les modèles de confiance à base de certification* [6, 23, 2, 24, 25, 1, 26, 27, 28]. Dans la première catégorie, la confiance est basée sur la notion de réputation. La réputation d'un nœud augmente quand il effectue correctement les tâches qui correspond au bon fonctionnement du réseau, tel que le routage. Chaque nœud observe le comportement de ses voisins et déclare une accusation s'il estime qu'un nœud est suspect, ce qui permet d'isoler l'ensemble des nœuds malicieux. Dans le cadre cet article, nous nous intéressons à la deuxième catégorie : *les modèles de confiance à base de certification*. En effet, un certificat, est une structure de données dans laquelle une clé est liée à une identité (et éventuellement à certains autres attributs) délivré par une tierce partie de confiance. Si

*. Université de A/Mira, Département d'Informatique, Béjaia, Algérie.
†. Université de Technologie de Compiègne, Heudiasyc-UMR CNRS 6599, Compiègne, France.

cette dernière estime qu'un nœud donné est digne de confiance, elle le délivre un certificat qui va lui permettre de prouver sa légitimité envers les autres nœuds du réseau.

Dans cet article, nous proposons une architecture d'un modèle de confiance à base de certification qui permet la coexistence de plusieurs autorités de certification hétérogènes. La solution proposée est partiellement distribuée, où le rôle de chaque autorité de certification est assuré par un ensemble de serveurs qui signent collectivement les certificats. Chaque autorité de certification supervise un ensemble d'utilisateurs appartenant à un domaine. Le passage d'un domaine à un autre se fait à l'aide d'un graphe de confiance établi entre les différentes autorités de certification.

Le reste de cet article est organisé comme suit. Dans la section 2, nous donnons une description technique du mécanisme de signature utilisé pour la délivrance des certificats. Dans la section 3, nous présentons notre modèle de confiance. Dans la section 4, nous donnons les résultats de simulations. Enfin, la section 5 conclut l'article.

2 La multisignature à base d'identité

2.1 Le schéma de multisignature de Harn et Ren

Terme	**Description**
PKG	*Private-Key Generator.*
d/e	La clé privée/publique du PKG.
id_i	L'identité de l'entité i.
$\mathcal{K}_i^{-1}$	La clé privée du signataire id_i.
$\mathcal{M}$	Un message.
$\mathcal{S}_i$	Une signature générée par le signataire id_i.
$\mathcal{H}$	Une fonction de hachage à sens unique.
σ	La multisignature.
k	Le nombre de signataires.

Table 1: Notations

Le schéma de multisignature de Harn et Ren est basé sur le schéma de signature de Shamir [14]. Le schéma permet à un ensemble de signataires de signer collectivement un seul message, où la vérification de la validité du message nécessite seulement les identificateurs des signataires. Le schéma suppose l'existence d'un serveur central de prédistribution de clés, nommé PKG (*Private-Key Generator*). Ce dernier, à l'aide de sa clé privée, génère pour chaque signataire une clé privée qui correspond à l'identité de ce dernier. Le schéma de multisignature de Harn et Ren s'exécute en trois étapes (cf. tableau 1 pour la terminologie la plus utilisée) :

1. *La génération de la clé privée du signataire*
   (a) Le signataire id_i soumet son identité au PKG.
   (b) Le PKG chiffre avec sa propre clé privée la valeur de id_i pour générer la clé privée $\mathcal{K}_i^{-1}$, tel que $\mathcal{K}_i^{-1} = id_i^d \bmod n$ ($\mathcal{K}_i^{-1}$ est la clé privée du signataire id_i).

2. *La génération de la multisignature*

   (a) Le signataire id_i choisit un nombre aléatoire r_i.

   (b) Le signataire id_i diffuse r_i à tous les autres signataires.

   (c) Lors de la réception de toutes les valeurs de r_j $(j = 1, 2, ..., k)$, le signataire id_i calcule :
   $t = \prod_{j=1}^{k} r_j \bmod n$ et $\mathcal{S}_i = \mathcal{K}_i^{-1}.r_i^{\mathcal{H}(t,\mathcal{M})} \bmod n$.

   (d) Le signataire id_i diffuse $\mathcal{S}_i$ à tous les autres signataires.

   (e) Lors de la réception de toutes les signatures $\mathcal{S}_j$ $(j = 1, 2, ..., k)$, le signataire id_i, calcule la multisignature $\mathcal{S}$: $\mathcal{S} = \prod_{j=1}^{k} \mathcal{S}_j \bmod n$. La multisignature du message $\mathcal{M}$ est $\sigma = (t, \mathcal{S})$.

3. *La vérification de la multisignature*
   Pour vérifier la multisignature $(\mathcal{M}, \sigma)$ des signataires $id_1, id_2, ..., id_k$, on vérifie l'égalité suivante : $\mathcal{S}^e \bmod n = \prod_{j=1}^{k} id_j.t^{\mathcal{H}(t,\mathcal{M})} \bmod n$.

Etape	**Action**	**Connaissances de id_j**
$1:a$ et $1:b$	id_j obtient sa propre clé privée $\mathcal{K}_j^{-1}$	$\mathcal{K}_j^{-1}$
$2:a$	id_j choisit r_j	r_j, $\mathcal{M}$, n, $\mathcal{K}_j^{-1}$
$2:b$	id_j diffuse r_j et reçoit tous les r_i : $r_j \longrightarrow$ $\longleftarrow r_1$ $\longleftarrow r_2$ ⋮ $\longleftarrow r_k$	$\{r_1, r_2, ..., r_k\}$, $\mathcal{M}$, n, $\mathcal{K}_j^{-1}$
$2:c$	id_j calcule t et $\mathcal{S}_j$	t, $\mathcal{S}_j$, $\{r_1, r_2, ..., r_k\}$, $\mathcal{M}$, n, $\mathcal{K}_j^{-1}$
$2:d$	id_j diffuse $\mathcal{S}_j$ et reçoit tous les $\mathcal{S}_i$: $\mathcal{S}_j \longrightarrow$ $\longleftarrow \mathcal{S}_1$ $\longleftarrow \mathcal{S}_2$ ⋮ $\longleftarrow \mathcal{S}_k$	$\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}$, t, $\{r_1, r_2, ..., r_k\}$, $\mathcal{M}$, n, $\mathcal{K}_j^{-1}$
phase d'attaque	Pour chaque signataire id_i, id_j peut mettre $a = r_i^{\mathcal{H}(t,\mathcal{M})}$ et $b = \mathcal{S}_i$ et résoudre l'équation $a.\mathcal{K}_i^{-1} \bmod n = b$ (cf. la formule de signature), et ainsi il calcule la valeur de toutes les clés $\mathcal{K}_i^{-1}$.	$\{\mathcal{K}_1^{-1}, \mathcal{K}_2^{-1}, ..., \mathcal{K}_k^{-1}\}$, $\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}$, t, $\{r_1, r_2, ..., r_k\}$, $\mathcal{M}$, n, $\mathcal{K}_j^{-1}$

Table 2: L'attaque de reconstitution de la clé privée

2.2 La vulnérabilité du schéma de Harn et Ren

En effet, le schéma de multisignature de Harn et Ren est vulnérable en cas de la présence de signataires malveillants. Un signataire interne id_j peut reconstituer la clé privée de

n'importe quel autre signataire id_i à l'étape de la génération de la multisignature. Pour démontrer ça, nous exécutons, étape par étape, le schéma de multsignature de Harn et Ren pour un signataire interne id_j (cf. tableau 2). Lors de l'exécution des étapes $1:a$ et $1:b$, le signataire id_j obtient sa propre clé privée $\mathcal{K}_j^{-1}$. A l'étape $2:a$, le signataire id_j choisit un nombre aléatoire r_j. A l'étape $2:b$, le signataire id_j diffuse r_j vers tous les autres signataires, et ainsi, il reçoit également tous les r_i ($\{r_1, r_2, ..., r_k\}$) diffusés par les autres signataires. A l'étape $2:c$, le signataire id_j calcule t et $\mathcal{S}_j$. A l'étape $2:d$, le signataire id_j diffuse $\mathcal{S}_j$ vers tous les autres signataires, et ainsi, il reçoit également toutes les signatures $\mathcal{S}_i$ ($\{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}$) diffusées par les autres signataires. A partir de là, le signataire id_j dispose de : $\{t, \mathcal{M}, n, \{r_1, r_2, ..., r_k\}, \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}, \mathcal{K}_j^{-1}\}$. Ainsi, le signataire id_j peut reconstruire la clé privée $\mathcal{K}_i^{-1}$ de n'importe quel signataire id_i en utilisant la formule de signature : $\mathcal{S}_i = \mathcal{K}_i^{-1}.r_i^{\mathcal{H}(t,\mathcal{M})} \bmod n$. Pour ce faire, le signataire id_j résout cette équation, qui est de la forme $ax \bmod n = b$, où les valeurs de $a = r_i^{\mathcal{H}(t,\mathcal{M})}$ et $b = \mathcal{S}_i$ sont connues. Pour résoudre ce type d'équation, on utilise l'algorithme étendu d'Eclid décri dans [11].

2.3 Notre schéma amélioré de multisignature

Pour contrer l'attaque présentée dans la Section 2.2, nous avons jugé qu'il est nécessaire de préserver la valeur de r_i secrète, pour objectif de rendre impossible la résolution de l'équation. Dans ce contexte, au lieu de soumettre r_i publiquement, nous proposons que le signataire le chiffre tout d'abord avec la clé publique du PKG, $\mathcal{R}_i = r_i^e \bmod n$, et diffuse $\mathcal{R}_i$ vers tous les autres des signataires en préservant r_i secrète à son niveau. Lors de la réception de toutes les valeurs de $\mathcal{R}_j$ $(j = 1, 2, ..., k)$, le signataire id_i calcule le composant $\mathcal{T} = \prod_{j=1}^{k} \mathcal{R}_j \bmod n$, et ensuite il calcule la signature $\mathcal{S}_i = \mathcal{K}_i^{-1}.r_i^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n$. Maintenant, nous pouvons constater qu'il est impossible de reconstituer r_i à partir de $\mathcal{R}_i$, car elle est chiffrer à l'aide de la clé publique du PKG. Enfin, la formule de la vérification de la multisignature devient :

$$\begin{aligned}
&\mathcal{S}^e \bmod n = \prod_{j=1}^{k} \mathcal{S}_j^e \bmod n \\
&= \prod_{j=1}^{k} (\mathcal{K}_j^{-1^e}).(r_j^e)^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n \\
&= \prod_{j=1}^{k} id_j.\mathcal{T}^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n
\end{aligned}$$

Pour vérifier la validité de la multisignature d'un message, on vérifie l'égalité $\mathcal{M}$, l'égalité $\mathcal{S}^e \bmod n = \prod_{j=1}^{k} id_j.\mathcal{T}^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n$. Dans ce qui suit, nous présentons notre schéma amélioré de multisignature :

1. *La génération de la clé privée du signataire*
   (a) Le signataire id_i soumet son identité au PKG.
   (b) Le PKG chiffre avec sa propre clé privée la valeur de id_i pour générer la clé privée $\mathcal{K}_i^{-1}$, tel que $\mathcal{K}_i^{-1} = id_i^d \bmod n$ ($\mathcal{K}_i^{-1}$ est la clé privée du signataire id_i).
2. *La génération de la multisignature*
   (a) Le signataire id_i choisit un nombre aléatoire r_i.
   (b) Le signataire id_i calcule $\mathcal{R}_i = r_i^e \bmod n$.
   (c) Le signataire id_i diffuse $\mathcal{R}_i$ à tous les autres signataires.
   (d) Lors de la réception de toutes les valeurs de $\mathcal{R}_j$ $(j = 1, 2, ..., k)$, le signataire id_i calcule :
   $\mathcal{T} = \prod_{j=1}^{k} \mathcal{R}_j \bmod n$, et $\mathcal{S}_i = \mathcal{K}_i^{-1}.r_i^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n$.

(e) Le signataire id_i diffuse $\mathcal{S}_i$ à tous les autres signataires.

(f) Lors de la réception de toutes les signatures $\mathcal{S}_j$ $(j = 1, 2, ..., k)$, le signataire id_i, calcule la multisignature $\mathcal{S}$: $\mathcal{S} = \prod_{j=1}^{k} \mathcal{S}_j \bmod n$. La multisignature du message $\mathcal{M}$ est $\sigma = (\mathcal{T}, \mathcal{S})$.

3. *La vérification de la multisignature*
Pour vérifier la multisignature $(\mathcal{M}, \sigma)$ des signataires $id_1, id_2, ..., id_k$, on vérifie l'égalité suivante : $\mathcal{S}^e \bmod n = \prod_{j=1}^{k} id_j.\mathcal{T}^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n$.

Dans notre schéma amélioré de multisignature, il est impossible pour n'importe quel signataire de reconstituer la clé privée de n'importe quel autre signataire. En effet, lors de l'exécution de notre schéma amélioré de multisignature, dans le pire des cas, un signataire malveillant est capable de récolter $\{\mathcal{T}, \mathcal{M}, n, \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_k\}, \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_k\}\}$. Selon notre formule de signature, $\mathcal{S}_i = \mathcal{K}_i^{-1}.r_i^{\mathcal{H}(\mathcal{T},\mathcal{M})} \bmod n$, le signataire est en face d'une équation sous forme : $b = x.y^a \bmod n$ (x et y ce sont les variables, $a = \mathcal{H}(\mathcal{T}, \mathcal{M})$ et $b = \mathcal{S}_i$), où il est impossible de calculer x et y à travers un algorithme polynomial.

2.4 Exemple

Dans cette sous-section, nous donnons un exemple illustratif de l'exécution de notre protocole de multisignature pour $k = 3$ signataires qui portent respectivement les identités 1, 2, et 3. La multisignature se fait pour $\mathcal{M} = 6$, $(\mathcal{K}_{PKG}^{-1}, \mathcal{K}_{PKG}) = (131, 11)$, $n = 527$, et une fonction de hachage simple définie comme suit : $\mathcal{H}(\mathcal{X}, \mathcal{Y}) = \mathcal{X}.\mathcal{Y}$.

- Génération de la clé privée de chaque signataire : $\mathcal{K}_1^{-1} = 1^{131} \bmod 527 = 1$, $\mathcal{K}_2^{-1} = 2^{131} \bmod 527 = 467$, et $\mathcal{K}_3^{-1} = 3^{131} \bmod 527 = 44$.
- Génération de la multisignature
  - Chaque signataire génère un nombre aléatoire r_i : $r_1 = 66$, $r_2 = 17$, et $r_3 = 90$.
  - Chaque signataire calcule $\mathcal{R}_i$: $\mathcal{R}_1 = 66^{11} \bmod 527 = 128$, $\mathcal{R}_2 = 17^{11} \bmod 527 = 425$, et $\mathcal{R}_3 = 90^{11} \bmod 527 = 266$.
  - $\mathcal{T} = 128.425.266 \bmod 527 = 34$.
  - $\mathcal{H}(\mathcal{T}, \mathcal{M}) = 34.6 = 204$.
  - Chaque signataire calcule une signature $\mathcal{S}_i$: $\mathcal{S}_1 = 1.66^{204} \bmod 527 = 101$, $\mathcal{S}_2 = 467.17^{204} \bmod 527 = 442$, et $\mathcal{S}_3 = 44.90^{204} \bmod 527 = 57$.
  - $\mathcal{S} = 101.442.57 \bmod 527 = 238$.
- Vérification de la multisignature : $1.2.3.34^{204} \bmod 527 = 136 = 238^{11} \bmod 527$.

3 Architecture de notre modèle de confiance

3.1 Description de notre modèle de confiance

Notre modèle comporte plusieurs autorités de certification hétérogènes (l'architecture globale de notre modèle de confiance est illustrée dans figure 1). Chaque autorité de certification est distribuée à travers un ensemble de serveurs qui signent des certificats pour les utilisateurs en utilisant notre schéma de multisignature. Nous supposons à travers notre modèle l'existence de plusieurs domaines, où chaque domaine est supervisé par une autorité de certification qui exécute sa propre politique de certification pour la délivrance des certificats aux utilisateurs appartenant à son domaine. Ainsi, l'authentification des

utilisateurs du même domaine se fait directement à travers l'autorité de certification locale. Dans ce modèle, nous supposons la préexistence de certaines relations de confiance entre les autorités de certification de chaque domaine, ce qui forme un graphe de confiance inter-autorités de certification. Ces relations de confiance sont nécessaires pour permettre à deux utilisateurs appartenant à deux domaines différents puissent s'authentifier.

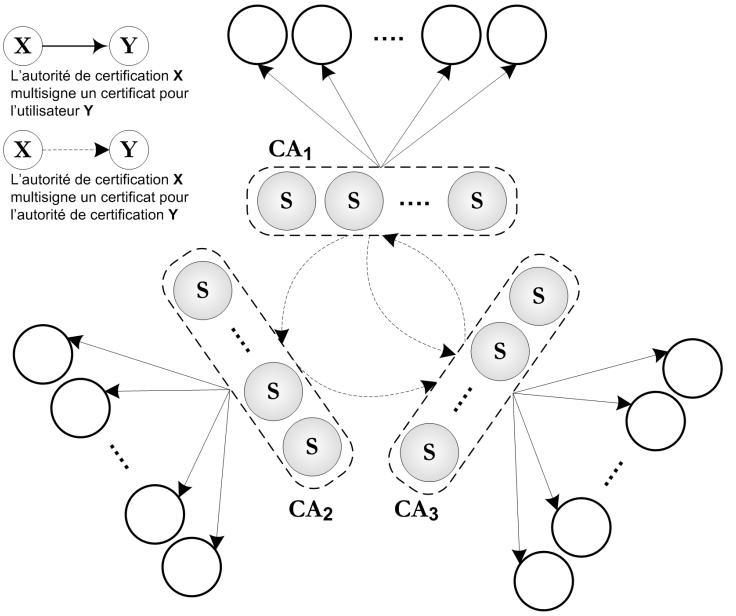


Figure 1: Notre modèle de confiance

3.2 Phase d'initialisation

Nous supposons l'existence d'un serveur central de prédistribution de clés qui va agir en tant que PKG. Ce serveur dispose d'une paire de clés $(\mathcal{K}^{-1}, \mathcal{K})$ qui représentent respectivement sa clé privée et publique. Pratiquement, ce serveur pourrait être un administrateur. Son rôle consiste à préconfigurer le système par la mise en œuvre d'un ensemble d'autorités de certification. Chacune d'elles sera représentée par un ensemble de serveurs qui portent des identités fixes et publiques. Chaque autorité de certification reçoit un certificat de rôle signé par le serveur central justifiant l'appartenance de l'ensemble des serveurs au domaine. De ce fait, chaque serveur peut prouver son rôle en tant qu'une partie de l'autorité de certification pour les utilisateurs. Egalement, le serveur central configure chaque serveur i par une clé privée $\mathcal{K}_i^{-1}$ qui correspond à son identité $i = \mathcal{K}_i$.

3.3 Délivrance des certificats

Chaque autorité de certification comporte un serveur délégué. Ce serveur doit être choisi par accord à travers l'ensemble des autres serveurs. Une fois élu, l'ensemble des serveurs (y compris lui-même) lui délivre un certificat de délégation à travers la mutisignature de l'ensemble des serveurs. Le serveur délégué est le représentant de l'autorité

de certification à l'intérieur et à l'extérieur du domaine. Ceci, ne veut par dire qu'il s'en charge de prendre les décisions de certification. Il est considéré comme une interface de négociation, ce qui permet d'éviter la diffusion des messages à la totalité des serveurs. S'il s'agit d'une requête de certification interne (le cas d'un utilisateur appartenant au même domaine), le serveur délégué va jouer le rôle de coordinateur en déclenchant le protocole de mutisignature pour délivrer le certificat à l'utilisateur. Ce dernier peut vérifier la validité du certificat seulement à travers les identités des serveurs. Dans le cas d'une requête de certification externe (inter-domaines ; une autorité de certification externe souhaite être certifiée par l'autorité de certification), le protocole de certification s'exécute en deux étapes : (1) *étape de négociation*, et (2) *étape de certification*.

3.3.1 Etape de négociation

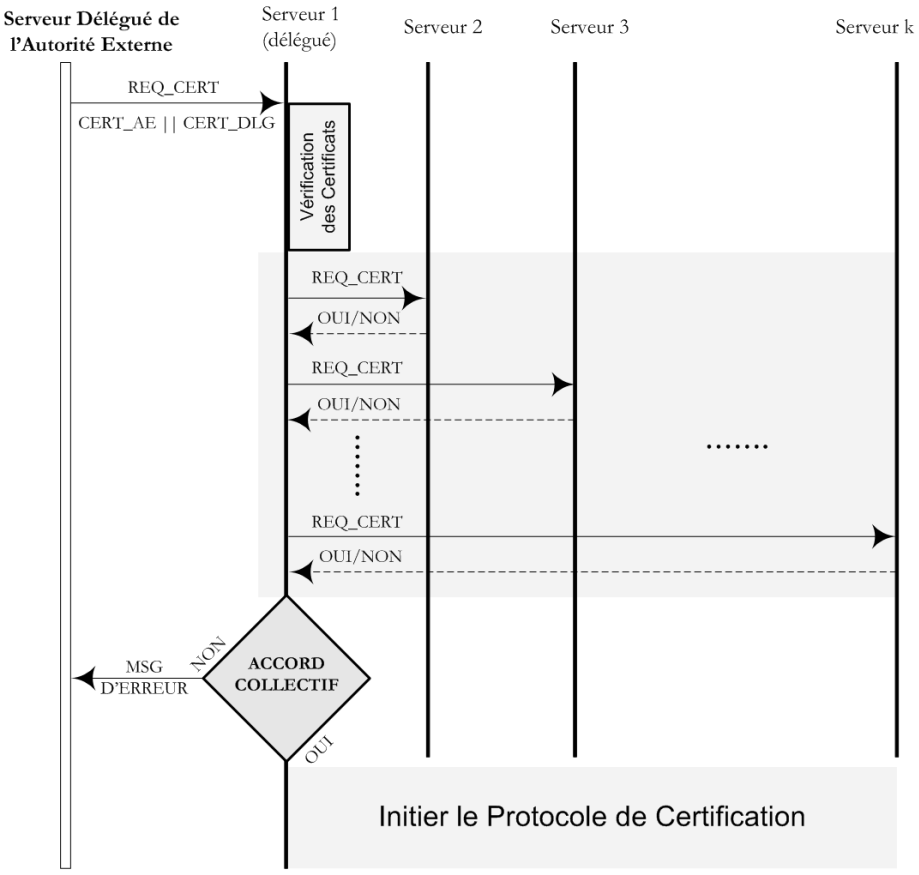
Le déroulement du protocole de négociation est illustré sur la figure 2. La requête de certification inter-domaines contient : (1) le certificat de rôle CERT_AE de l'ensemble des serveurs appartenant à l'autorité externe, et (2) le certificat de délégation CERT_DLG du serveur délégué. En recevant la requête, le serveur délégué vérifie la validité du certificat CERT_CA en utilisant la clé publique $\mathcal{K}$ du serveur central. Egalement, il vérifie la multisiagnature du certificat de délégation vis-à-vis les identités des serveurs figurés dans le certificat de rôle. Ensuite, il diffuse la requête à l'ensemble des serveurs pour déclencher la procédure de négociation. Chaque serveur en recevant la requête, s'il estime que l'autorité externe est digne de confiance, il envoie son accord au serveur délégué. Si ce dernier, reçoit un accord collectif, il prépare la procédure de certification en générant un certificat pour l'autorité de certification externe. Ce certificat sera diffusé à l'ensemble des serveurs pour l'étape de la multisignature. Si l'accord est défavorable, un message d'erreur lui sera transmis.

3.3.2 Etape de certification

Le déroulement du protocole de certification est illustré sur la figure 3 (le protocole de certification est basé sur le protocole de multisignature décrit dans la section 2.3). En recevant le certificat, chaque serveur i génère un nombre aléatoire $\mathcal{R}_i$, le chiffre avec la clé publique du serveur central et l'envoie au serveur délégué. Ce dernier, en recevant toutes les valeurs de $\mathcal{R}_i$, il calcule $\mathcal{T}$ et le diffuse à l'ensemble des serveurs. Ensuite, chaque serveur calcule la signature $\mathcal{S}_i$. A la fin, le serveur délégué calcule la multisignature $\mathcal{S}$ du certificat, et il le renvoie au serveur délégué de l'autorité externe.

3.4 Authentification des clés publiques

L'authentification des clés publique se fait à travers la vérification de la validité des certificats. S'il s'agit d'une communication interne, chaque utilisateur vérifie le certificat de son interlocuteur vu qu'ils sont supervisés par la même autorité de certification. Si les deux utilisateurs appartiennent à deux domaines différents, les deux utilisateurs doivent vérifier l'existence d'une relation de confiance mutuelle entre leurs autorités de certification auxquelles ils appartiennent. De ce fait, chaque utilisateur essaye de trouver une chaîne de certificats inter-domaines qui relie son autorité de certification avec celle de son interlocuteur. Si une telle chaîne est trouvée, il vérifie la validité des multisignatures générées pour chaque certificat. S'il existe au moins une chaîne de certificats valide, l'authentification est établie.



REQ_CERT : Requête de Certification.
CERT_AE : Certificat qui Prouve le Rôle des Serveurs de l'Autorité Externe.
CERT_DLG : Certificat de Délégation du Serveur Délégué.

Figure 2: Phase de négociation

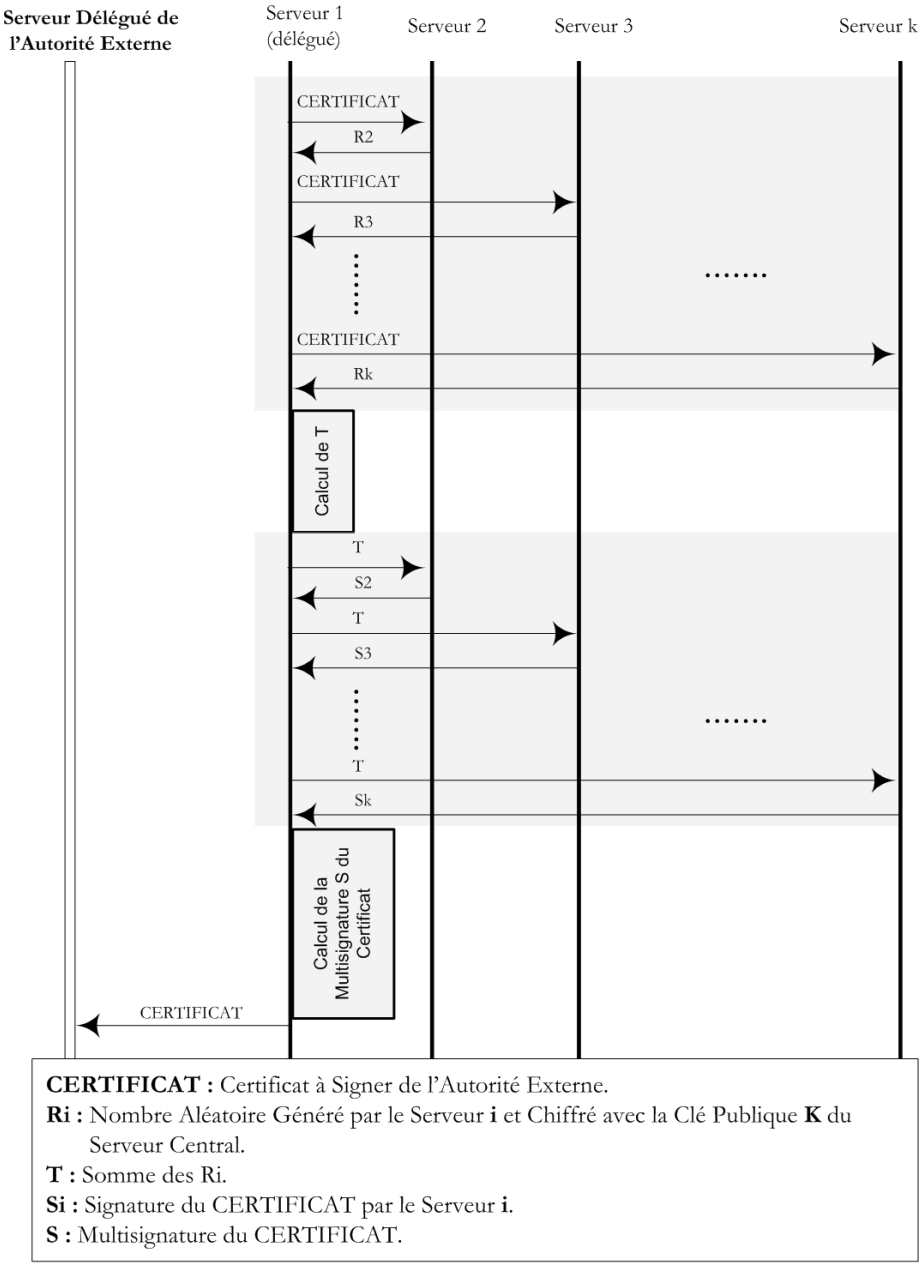


Figure 3: Phase de certification

4 Résultats de simulations

4.1 Environnement et paramètres de simulations

Les simulations sont faites sous l'environnement *matlab*. Nous avons opté pour une durée de simulation de 3600 s. Les requêtes des utilisateurs arrivent aux autorités de certification selon une loi de Poisson avec une durée moyenne de 10 s. Le simulateur estime si un lien radio existe entre deux nœuds quelconques en fonction de la distance qui les sépare. Les nœuds se déplacent sur une surface rectangulaire de 1000 m^2. Le déplacement des nœuds suit le modèle de mobilité *random waypoint* [29] avec une vitesse variable entre 0 et 20 m/s et une durée de pause variable entre 0 et 20 s. Les nœuds ont les mêmes caractéristiques matérielles et la même puissance de traitement. Le graphe de confiance inter-domaines est fixé par le simulateur d'une manière aléatoire. Pour chaque requête, nous tirons deux nœuds appartenant à deux domaines différents d'une manière aléatoire, à travers laquelle les deux nœuds tentent de récolter une chaîne de certificats reliant les deux domaines. Le critère évalué, à travers cette simulation, est le taux moyen de certificats réussis. Les impacts étudiés sont respectivement : le nombre d'autorités de certification et nombre de serveurs de certification.

4.2 Impact du nombre d'autorités de certification

Dans cette sous-section, nous étudions l'impact du nombre d'autorités de certification (noté n) sur le taux moyen de certificats réussis. Nous avons effectué cette simulation pour une variation de valeurs de $n = 1$ à $n = 10$ autorités de certification pour trois cas de portées de communication : $p = 50$ m, $p = 100$ m, et $p = 150$ m. Chaque autorité de certification comporte $k = 5$ serveurs qui supervisent un certain nombre de nœuds. Les résultats de cette simulation sont illustrés sur la figure 4. Nous constatons que le nombre de certificats réussis se dégrade légèrement durant la variation de n. Cela veut dire, que le nombre d'autorités de certification n'a pas un impact très fort sur le taux moyen de certificats réussis. En effet, quand nous augmentons n, le réseau subit un partitionnement de plusieurs domaines de certification. Si les deux nœuds appartiennent au même domaine, il suffit seulement de vérifier les certificats de l'un de l'autre vu que l'autorité de certification est commune pour les deux nœuds. Malgré ça, cette vérification peut s'échouer si les serveurs de certification ne sont pas accessibles lors de la requête. Si les deux nœuds appartiennent à deux domaines différents, la réussite de certification dépend de l'existence d'une chaîne de certificats reliant les deux domaines. Pour un nombre réduit d'autorités de certification, la probabilité que les nœuds soient du même domaine est grande contrairement au cas d'un nombre important d'autorités de certification. Ceci interprète la dégradation du taux moyen de certificats réussit.

4.3 Impact du nombre de serveurs k

Dans cette sous-section, nous étudions l'impact du paramètre k : le nombre de serveurs impliqués dans chaque autorité de certification. Nous avons effectué cette simulation pour une variation de valeurs de $k = 5$ à $k = 20$ serveurs pour trois cas de portées de communication : $p = 50$ m, $p = 100$ m, $p = 150$ m et un nombre de $n = 5$ autorités de certification. Les résultats de cette simulation sont illustrés sur la figure 5. Nous constatons que le nombre de certificats réussis se dégrade considérablement durant la variation de k. En effet, à chaque requête, la totalité des k serveurs est sollicité, ce qui fait que le

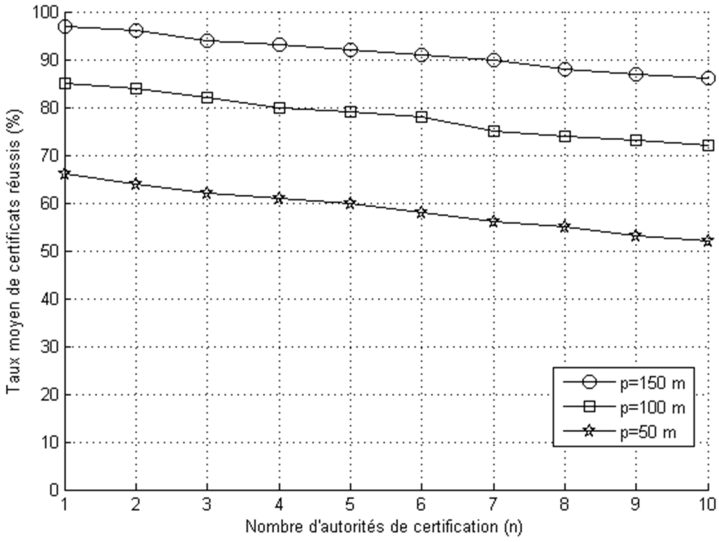


Figure 4: Impact du nombre d'autorités de certification sur le taux moyen de certificats réussis. $k = 5$ serveurs

taux moyen de certificats réussis est fortement sensible à la grandeur de k. Si ce dernier est grand, ceci rend difficile d'avoir tous les k serveurs accessibles à un moment donné vu la nature du réseau ad hoc mobile. De ce fait, ce paramètre doit être ajusté réduit afin d'améliorer la disponibilité du service de certification.

5 Conclusion

Dans cet article, nous avons proposé un modèle de confiance à base de certification qui prend en compte l'hétérogénéité des autorités de certification. Le service de certification dans chaque domaine repose sur l'aspect collaboratif avec une gestion partiellement distribuée des certificats à travers un ensemble de serveurs. Notre modèle fait l'objet d'une solution hybride à travers laquelle la certification est autoritaire à l'intérieur de chaque domaine, et elle est anarchique inter-domaine. Cette solution élimine la centralisation du service de certification par une unique autorité de certification, ce qui la rend la solution très pratique. Pour mettre en valeur les qualités de performance de notre modèle, nous avons effectué des simulations, où ces dernières ont montré que notre modèle n'est pas fortement influencé par le nombre d'autorité de certification dans un milieu ad hoc. Cependant, il est nécessaire de réduire le nombre de serveurs impliqués dans chaque autorité de certification afin d'aboutir à un taux acceptable de certificat réussis.

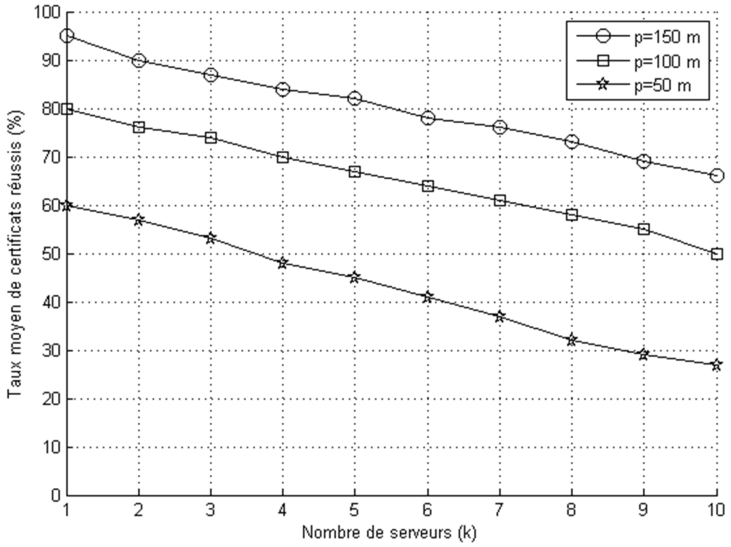


Figure 5: Impact du nombre de serveurs k. $n = 5$ autorités de certification

Références

[1] J. Luo, J. Hubaux, P. Eugster. *DICTATE : Distributed Certification Authority with Probabilistic Freshness for Ad hoc Networks*. IEEE Transactions on Dependable and Secure Computing, 2005.

[2] S. Capkun, L. Buttyan, J. Hubaux. *Self-ogranized public key management for mobile Ad hoc networks*. IEEE Transactions on Mobile Computing, 2003.

[3] L. Zhou, F. Schneider, R. Renesse. *COCA : A Secure Distributed Online Certification Authority*. ACM Transactions Computing Systems, 2002.

[4] H. Luo, S. Lu. *Ubiquitous and Robust Authentication Services for Ad hoc Wireless Networks*. Technical Report, UCLA Computer Science, 2000.

[5] R. Perlman. *An Overview of PKI Trusts Models*. IEEE Network, 1999.

[6] L. Zhou, Z. Haas. *Securing Adhoc Networks*. IEEE Network, 1999.

[7] A. Abdulrahman. *The PGP Trust Model*. The Journal of Electronic Commerce, 1997.

[8] A. Abdulrahman, S. Hailes. *A Distributed Trust Model*. In Proceedings 97 New Security Paradigms, 1997.

[9] J. Kohl, B. Neuman. *The Kerberos Network Authentication Service Version 5*. RFC-1510, 1991.

[10] C. Bettstetter. *Topology Properties of Ad Hoc Networks with Random Waypoint Mobility*. In Proceedings of ACM Int. Symposium in Mobile Ad Hoc on Networking and Computing (MobiHoc), 2003.

[11] M. Bishop. *Computer Security : Art and Science*. Addison Wesley Professional, 2004.

[12] N. Demytko. *A new elliptic curve based analogue of RSA*. Lecture Notes in Computer Science, Springer-Verlag, 1994.

[13] N. Koblitz. *Elliptic curve cryptosystems*. Mathematics of Computation, 1987.

[14] A. Shamir. *Identity-based cryptosystems and signature schemes*. Lecture Notes in Computer Science, Berlin, Springer-Verlag, 1985.

[15] ITU-T Recommendation X509/ISO/IEC 9594-8. *Public-Key and Attribute Certificate Frameworks*. 4th edition, 2001.

[16] S. Buchegger, J.Y. Le-Boudec. *Performance analysis of the CONFIDANT protocol*. In Proceedings of 3rd ACM International Symposium, on Mobile Ad Hoc Networking and Computing, 2002.

[17] P. Michiardi, R. Molva. *CORE : a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks*. In Proceedings of 6th IFIP Communication and Multimedia Security Conference, 2002.

[18] H. Janzadeh, K. Fayazbakhsh, M. Dehghan, M.S. Fallah. *A secure credit-based cooperation stimulating mechanism for MANETs using hash chains*. Future Generation Computer Systems, 2009.

[19] Q. He, D. Wu, P. Khosla. *SORI : a secure and objective reputation-based incentive scheme for ad-hoc networks*. In Proceedings of IEEE WCNC'04, 2004.

[20] C. Zouridaki, B.L. Mark, M. Hejmo, R.K. Thomas. *E-Hermes : A robust cooperative trust establishment scheme for mobile ad hoc networks*. Ad Hoc Networks, 2009.

[21] E. Ayday, F. Fekri. *A protocol for data availability in Mobile Ad-Hoc Networks in the presence of insider attacks*. Ad Hoc Networks, 2010.

[22] N. Marchanga, R. Dattab. *Collaborative techniques for intrusion detection in mobile ad-hoc networks*. Ad Hoc Networks, 2008.

[23] S. Capkun, L. Buttyan, J. Hubaux. *Small Worlds in Security Systems - an Analysis of the PGP Certificate Graph*. In Proceedings of New Security Paradigms Workshop (ACM), 2002.

[24] S. Raghani, D. Toshniwal, R. Joshi. *Dynamic Support for Distributed Certification Authority in Mobile Ad Hoc Networks*. In Proceedings International Conference on Hybrid Information Technology (IEEE), 2006.

[25] S. Yi, R. Kravets. *MOCA - Mobile Certificate Authority for Wireless Ad hoc Networks*. In Proceedings of the Second Annual PKI Research Workshop, 2003.

[26] M. Ge, K.Y. Lam, D. Gollmann, S.L. Chung, C.C. Chang, J.B. Li. *A robust certification service for highly dynamic MANET in emergency tasks*. Wiley InterScience : International Journal of Communication Systems, 2009.

[27] Y. Kitada, Y. Arakawa, K. Takemori, A. Watanabe, I. Sasase. *On demand distributed public key management using routing information for wirelss ad hoc netwoks*. IEICE Transactions on Information and Systems, 2005.

[28] G. Kambourakis, E. Konstantinou, A. Douma, M. Anagnostopoulos, G. Fotiadis. *Efficient Certification Path Discovery for MANET*. EURASIP Journal on Wireless Communications and Networking, 2010.

[29] C. Bettstetter. *Topology Properties of Ad Hoc Networks with Random Waypoint Mobility.* In Proceedings of ACM Int. Symposium in Mobile Ad Hoc on Networking and Computing (MobiHoc), 2003.