Méthodes de révocation de certificats

Etudes et comparaisons

Chakib Bekara[α]- Maryline Laurent-Maknavicius[α]

[α] *Laboratoire SAMOVAR, département Logiciel-Réseaux (LOR)*
Institut National des Télécommunications d'Evry

{Chakib.Bekara, Maryline.Maknavicius}@int-evry.fr

Résumé. Cet article présente un ensemble non exhaustif de méthodes de révocation de certificats classées en trois catégories : les schémas à structure de listes, les schémas à structure d'arbre et les schémas transparents aux vérificateurs. Chaque méthode est présentée avec les différents coûts induits, comme le coût en bande passante, le coût en calcul, la fenêtre de vulnérabilité, l'extensibilité,…Une analyse comparative de ces méthodes est donnée, le but étant d'aider les ingénieurs et les concepteurs dans le choix de la méthode en fonction de leurs besoins et des moyens dont ils disposent.

Abstract. This paper presents a subset of revocation methods classified into three categories: list-based schemas, tree-based schemas and verifier transparent schemas. Each method is presented along with inherent costs like bandwidth, processing cost, window of vulnerability, scalability,… A comparative analysis of those methods is given with the objective to help engineers and designers selecting a method according to their own needs and means.

Mots clés : *Certificat numérique, PKI, IGC, X.509, gestion de certificats, révocation, authentification, non répudiation*.

Keywords : Digital certificate, PKI, X.509, certificate management, revocation, authentication, non repudiation.

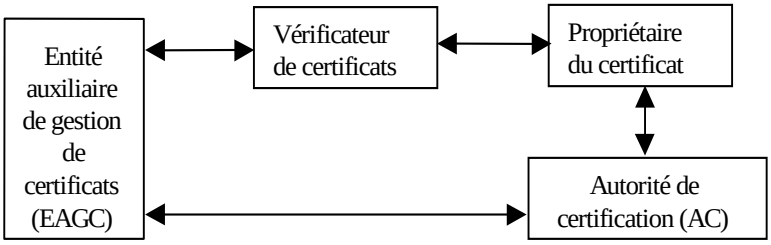
1 Introduction

L'usage des certificats de clés publiques s'est répandu ces dernières années sur Internet pour rendre plus sûres certaines applications comme les échanges de messages électroniques, les transactions électroniques, … Un certificat électronique peut en fait être assimilé à une carte d'identité associant une clé publique à son propriétaire ; une durée de validité la caractérise et son authenticité est apportée par la signature apposée par une autorité de certification. La gestion et l'utilisation des certificats sont réglementées par les infrastructures de gestion de clés (PKI - *Public Key Infrastructure*) qui comprennent entre autres des autorités de certification qui sont des tierces partis de confiance (TPCs), des serveurs de publication… (cf. Figure 1).

Si les certificats sont devenus d'usage courant, l'opération de révocation des certificats n'est pas encore au point. Pourtant, c'est elle qui permet d'invalider un certificat avant même que sa date d'expiration ne soit atteinte, et ce, pour différentes raisons : la compromission de la clé privée, le changement d'informations du certificat… L'opération de révocation consiste en fait à rendre public l'état d'un certificat pour que des utilisateurs en prennent connaissance, et par conséquent refusent tout document signé ou chiffré avec la clé privée révoquée, associée au certificat révoqué.

Plusieurs méthodes de révocation ont été proposées pour répondre à la diversité en besoin de sécurité et en capacité de ressources (calcul, stockage et bande passante) des applications et des utilisateurs. Dans [Arn00A] [Arn00B] [Zhe03], trois catégories de méthodes de révocation sont définies : les schémas de révocation à structure de listes (*List-based*), les schémas de révocation à structure d'arbres (*Tree-based*), et les schémas transparents aux vérificateurs (*Verifier-transparent*).

Cet article propose de décrire un ensemble représentatif de méthodes de révocation de ces trois catégories avec leurs caractéristiques et leurs coûts en bande passante et en calcul. Cette description fait l'objet des sections 3, 4 et 5 et s'appuie sur les critères proposés à la section 2 pour évaluer chacune des méthodes. Enfin la section 6 présente une analyse comparative de ces différentes méthodes. Noter que quelques études [Arn00A] [Arn00B] [Zhe03] ont permis d'établir une comparaison de ces méthodes, mais notre article permet d'étendre et compléter ces travaux.



Autorité de certification **(AC)** : Tierce Partie de Confiance **(TPC)** responsable de la création, du renouvellement, de la mise à jour et de la révocation des certificats électroniques

Vérificateur : entité désirant vérifier la validité d'un certificat

Entité Auxiliaire de Gestion de Certificats **(EAGC)** : entité permettant de renseigner un vérificateur quant à l'état de révocation d'un certificat. Cette entité peut être de confiance (serveur OCSP, serveur SEM) ou non (répertoire de publication)

Figure 1. Composants d'une PKI simplifiée [Zhe03]

2 Critères d'évaluation et notations

Plusieurs critères d'évaluation [Arn00A] [Arn00B] [Zhe03] peuvent être pris en considération afin de comparer les méthodes de révocation de certificats, à savoir :

Coût en mise à jour (**CMA** :*Update Cost*) : coût quotidien pour la mise à jour d'informations de révocation de certificats. Le coût se présente sous la forme d'un triplet (*bw-u*, $ovh_{AC\text{-}u}$, $ovh_{EAGC\text{-}u}$) présenté dans le Tableau 1.

Coût en requête de vérification (**CR** : *Query Cost*) : coût quotidien nécessaire à la validation d'un certificat (s'il est révoqué ou pas). Le coût est mesuré en fonction d'un quadruplet (*bw-q*, $ovhc_{AC\text{-}q}$ $ovh_{EAGC\text{-}q}$, $ovh_{verifier\text{-}q}$) (cf. Tableau 1).

Coût en mise à jour d'un certificat	*bw-u* : bande passante requise entre ACs et EAGCs $Ovh_{AC\text{-}u}$: coût de calcul au niveau de l'AC $Ovh_{EAGC\text{-}u}$: coût de calcul au niveau de l' EAGC
Coût en requête de vérification de certificat	*bw-q* : bande passante requise entre EAGCs et vérificateurs $Ovh_{AC\text{-}q}$: coût de calcul au niveau de l'AC $Ovh_{EAGC\text{-}q}$: coût de calcul au niveau d'une EAGC $Ovh_{verifier}$: coût de calcul au niveau des vérificateurs

Tableau 1. *Coûts en mise à jour et en requête de vérification (cf. figure 1 pour les abréviations)*

Fraîcheur des informations de révocation : degré de fraîcheur des informations de révocation. Il est important que les informations de révocation soient les plus fraîches possible.

Fenêtre de vulnérabilité (***WoV** : Window of Vulnerability*) : période durant laquelle le certificat peut être considéré valide alors qu'il est révoqué. Ce critère est fortement lié au critère précédent.

Extensibilité (*scalability*) : adaptabilité d'un schéma de révocation aux grands réseaux, en mesurant le rapport de l'accroissement des coûts en mise à jour, et en requête sur l'accroissement de la taille de la communauté desservie mesurée en nombre de certificats.

Risque système : risque qu'un système de révocation de certificat soit corrompu suite à une attaque. Le risque est d'autant plus élevé que les TPCs sont mis en ligne.

Déni de Service (*DoS*) : risque qu'un utilisateur du système de révocation de certificat ne puisse plus accéder aux informations de révocation, suite à une indisponibilité temporaire des ACs, répertoires de publications, serveurs de confiances, … Cette situation peut se produire lorsqu'une machine reçoit un grand volume d'informations (en provenance d'un client légitime ou non), et n'est plus capable à ce moment là de répondre aux requêtes des utilisateurs légitimes.

Schéma On-line/Off-line : fréquence de connexion d'un utilisateur aux EAGCs pour vérifier la révocation d'un certificat, un schéma On-line requérant beaucoup de connexions, et un schéma Off-line très peu.

Le Tableau 2 décrit les notations adoptées dans l'article pour représenter les différents coûts décrits ci-dessus.

Symbole	Description
n	Nombre de certificats dans la PKI (hors certificats révoqués)
p	Pourcentage de certificats révoqués par jour
q	Nombre de requêtes de demande d'état de révocation des certificats par jour
f	Fréquence de mises à jour des informations de révocation par jour
k	Nombre moyen de certificats gérés par une AC
$\frac{n}{k}$	Nombre moyen d'ACs dans la PKI
t	Durée de vie moyenne (validité moyenne) d'un certificat, mesurée en nombre de jours
l_{hash}	Longueur du résultat d'une fonction de hachage
l_{sig}	Longueur d'une signature électronique
l_{sn}	Longueur du numéro de série d'un certificat électronique
c_{hash}	Coût d'une opération de hachage (temps nécessaire)
c_{sign}	Coût d'une opération de génération d'une signature électronique
c_{verify}	Coût d'une opération de vérification d'une signature électronique

Tableau 2. *Notations*

3 Les schémas de révocation à structure de listes

Ces schémas adoptent la publication de listes pour la gestion des certificats révoqués.

3.1 La CRL [Arn00A] [Kik99] [Zhe03]

Une CRL *(Certificate Revocation List)* contient la liste des certificats révoqués, datée et signée par une AC et périodiquement publiée. Pour vérifier la validité d'un certificat, le vérificateur doit envoyer une requête au serveur de publication hébergeant la CRL correspondante, avec comme argument l'identifiant de l'AC en charge du certificat ; il reçoit alors la dernière CRL générée par l'AC ; il doit ensuite vérifier la signature de la CRL et sa durée de validité, et puis rechercher le certificat dans la CRL.

L'avantage de la CRL est sa simplicité, sa richesse en information et son faible risque système. Toutefois, la taille de la CRL constitue son inconvénient majeur, car la bande passante nécessaire à la mise à jour et à la vérification est très élevée, ce qui limite considérablement son extensibilité.

Pour garantir sa fraîcheur, la CRL contient la date de la prochaine mise à jour de la CRL. De ce fait, les vérificateurs qui ont besoin d'informations de révocation fraîches vont tous au même moment vouloir récupérer la nouvelle CRL. Cela risque donc de provoquer une implosion des requêtes CRLs.

Les coûts associés à la méthode sont donnés à la Figure 2.

Il existe d'autres variantes, qui sont des extensions et des améliorations de la méthode CRL. Dans ce qui suit, du fait de la longueur limitée de l'article, nous présentons une liste non exhaustive de ces

variantes sans les coûts associés. Pour une description plus détaillée, le lecteur intéressé pourra se reporter à [Bek04].

CMA : (f.(n.p. $\frac{t}{2}$ .l_{sn} + $\frac{n}{k}$ .l_{sig}), f.c_{sign}, f.c_{verify}) (a)

CR : (q.($k.p.\frac{t}{2}$ .l_{sn} + l_{sig}), 0, 0, $q.c_{verify}$) (b), tel que $\frac{t}{2}$ est la durée moyenne de mise en CRL d'un certificat révoqué.

WoV : $\frac{1}{f}$, c'est-à-dire la période de publication de la CRL.

Figure 2. Caractéristiques de la méthode CRL

3.1.1 Delta-CRL (Δ-CRL) [Arn00A] [Arn00B] [Kik99]

La Δ-CRL est une liste signée qui contient tous les certificats révoqués depuis la publication de la dernière CRL (CRL de base). Ainsi, la vérification d'un certificat va nécessiter de récupérer à la fois la CRL de base et la Δ-CRL la plus récente.

La Δ-CRL améliore la fraîcheur des informations de révocation par rapport aux CRLs, ce qui la rend plus extensible. Cependant, les coûts induits sont augmentés, et le problème d'implosion des requêtes persiste toujours. Enfin, la Δ-CRL présente un faible risque système.

3.1.2 CRL Distribution Points (CRL DP) [Arn00A] [Arn00B] [Kik99]

Cette méthode consiste à diviser une CRL en segments contenant chacun un sous-ensemble des certificats révoqués par une AC. Comme tout certificat contient un pointeur vers le segment lui correspondant, un vérificateur va pouvoir accéder directement au segment concerné. Ainsi la méthode CRL DP apparaît plus extensible que la CRL, et tout comme la CRL, elle présente un faible risque système.

3.1.3 Over-issued CRL [Arn00A] [Arn00B] [Kik99]

Cette méthode réduit considérablement l'implosion des requêtes CRLs, en permettant de délivrer des CRLs de même durée de vie qui se chevauchent dans le temps. De cette façon, les CRLs n'expirent pas en même temps au niveau des vérificateurs. Cette méthode augmente le coût en mise à jour, car les CRLs délivrées sont plus nombreuses. Par contre, la fenêtre de vulnérabilité reste inchangée et le risque système de la méthode est faible.

4 Les schémas de révocation à structure d'arbre

Dans ces schémas, les ACs ainsi que les répertoires de publication utilisent des structures d'arbres pour représenter les informations de révocation.

4.1 2-3 CRT (Certificate Revocation Tree)

La méthode 2-3 CRT [Mun04] [Nao00] [Zhe03] utilise un arbre d'ordre 2-3 dans lequel les feuilles correspondent aux certificats révoqués et sont ordonnées dans l'ordre croissant des numéros de série. La valeur d'un nœud de l'arbre est le hachage des valeurs de ses fils et la valeur racine est signée par l'AC pour garantir son authenticité.

Un certificat est considéré révoqué s'il apparaît comme une feuille de l'arbre; un certificat sera considéré valide si deux certificats correspondent à deux feuilles adjacentes de l'arbre avec pour l'un, un numéro de série supérieur au certificat recherché et pour l'autre un numéro inférieur.

L'arbre 2-3 CRT est mis à jour périodiquement par l'AC. Pour réduire la bande passante nécessaire à la mise à jour des serveurs de publication (cf. Figure 3), sont envoyés uniquement les certificats révoqués accompagnés d'un message daté et signé avec la clé privée de l'AC contenant la valeur racine de l'arbre, sa profondeur ainsi que sa durée de validité. Le serveur de publication vérifie la signature du message ainsi que sa durée de validité, puis met à jour sa copie de l'arbre, en vérifiant sa profondeur et sa valeur racine.

Pour vérifier un certificat, le vérificateur doit récupérer une branche (en cas de révocation) voire deux branches ainsi que la racine signée de l'arbre. Le vérificateur s'assure alors de l'authenticité de la signature sur la racine et en cas de certificat valide, il vérifie que les deux branches reçues, sont celles de deux certificats révoqués adjacents.

On peut considérer que la méthode 2-3 CRT est extensible au vu de la bande passante consommée (CMA et CR) et que le risque système est faible.

CMA : $(f.(\frac{n.p}{f}.l_{sn} + \frac{n}{k} l_{sig}),\ f.(k.p.log_3(k.p.\frac{t}{2}).c_{hash} + c_{sign}),\quad f.(k.p.log_3(k.p.\frac{t}{2}).c_{hash} + c_{verify}))$ (c)

CR : $(q.(2.log_3(p.k.\frac{t}{2})+1).l_{hash}+l_{sig}),\ 0,\ 0,\ q.(c_{verify}+log_3(p.k.\frac{t}{2}).c_{hash}))$ (d)

WoV : $\frac{1}{f}$

Figure 3. Caractéristiques de la méthode 2-3 CRT

5 Les schémas transparents aux vérificateurs *(verifier-transparent)*

Pour ces schémas, le vérificateur délègue la vérification du certificat auprès d'un serveur tiers.

5.1 OCSP (Online Certificate Status Protocol)

Cette méthode [rfc2560] [Zhe03] - qui est un standard de l'IETF (RFC2560) - introduit un serveur OCSP de confiance auprès duquel les vérificateurs vont s'adresser pour connaître l'état de validité d'un certificat ('bon', 'révoqué', 'inconnu'). Le vérificateur qui a connaissance du certificat du serveur doit vérifier l'authenticité des messages signés et retournés par le serveur. Les serveurs OCSP peuvent être dissociés des ACs, voire co-localisés avec l'AC, auquel cas l'AC et le serveur partagent les mêmes répertoires, ce qui garantit une fraîcheur optimale des informations de révocation (cf. Figure 4), mais induit un risque système important (car le serveur OCSP est en ligne).

Suivant le point de vue, cette méthode peut paraître extensible. En effet, elle est très consommatrice en calcul du fait que chaque requête de certificat parvenant au serveur suppose un message signé en retour. Par contre, elle consomme peu de bande passante du fait des messages courts échangés entre serveur OCSP et vérificateur.

CMA : négligeable
CR : $(q.l_{sig}, 0, q.c_{sig}, q.c_{verify})$ (e)

WoV ≈ 0

Figure 4. Caractéristiques de la méthode OCSP en cas de co-localisation de l'AC et du serveur OCSP

5.2 SEM (SEcurity Mediator scheme)

Cette méthode [Bon01] [Zhe03] est basée sur une variante du protocole **RSA**, appelée **mRSA** (***m****ediated* ***RSA***). Dans cette méthode c'est l'AC qui crée la paire de clés privée/publique des utilisateurs. La clé privée comprend deux parties distinctes, l'une détenue par l'utilisateur, l'autre détenue par un serveur de confiance en ligne appelé SEM.

Pour signer ou déchiffrer un message, l'utilisateur a besoin de coopérer avec le serveur SEM. Il doit lui envoyer le message à signer ou à déchiffrer ; le serveur SEM vérifie que le certificat de l'utilisateur n'est pas révoqué, calcule sa contribution en utilisant sa clé privée, et envoie le résultat à l'utilisateur qui va par la suite calculer sa propre contribution. Dans cette méthode, l'existence de la signature est une preuve de la validité du certificat au moment de la signature. Cette méthode suppose que l'AC envoie au moins quotidiennement au serveur SEM la liste des certificats révoqués.

Ainsi, toute la sécurité repose sur l'authenticité des serveurs SEM qui jouent en partie le rôle des vérificateurs. De ce fait, cette méthode présente un risque système élevé et est peu extensible du fait de la sollicitation importante du serveur SEM. Les coûts de la méthode SEM sont présentés dans la Figure.5

CMA : $(f.(\frac{n.p}{f}.l_{sn} + \frac{n}{k}.l_{sig}),\ f.c_{sign},\ f.c_{verify})$ (f)

CR : $(q.l_{sig}, 0, q.c_{sign}, 0)$ (g)

WoV : $\frac{1}{f}$

Figure 5. Caractéristiques de la méthode SEM

6 Comparaison des méthodes de révocation

À partir des formules (a) à (g), nous proposons de comparer les méthodes de révocation de certificats en fonction des critères d'évaluation présentés à la section 2.

Fenêtre de vulnérabilité (WoV)

La plupart des méthodes ont une fenêtre de vulnérabilité de $\frac{1}{f}$, qui représente la période de publication des informations de révocation (CRL, CRT, etc.). Pour la méthode Δ-CRL, la fenêtre WoV est plus petite car la liste des certificats révoqués est publiée plus fréquemment. Pour OCSP, elle peut avoisiner le zéro, ce qui met OCSP au premier rang des méthodes pour la fraîcheur des informations de révocation.

Coût en mise à jour (CMA)

Du fait du volume important des CRLs (certificats révoqués pouvant être publiés plusieurs années jusqu'à leur date d'expiration), la méthode CRL et ses dérivées sont gourmandes en bande passante lors d'une mise à jour des informations de révocation. L'approche OCSP est la moins gourmande car sa consommation peut être nulle en cas de co-localisation du serveur OCSP avec l'AC, puis viennent les approches 2-3 CRT et SEM.

Les méthodes CRL, SEM et OCSP représentent le coût en calcul au niveau de l'AC le plus bas (coût d'une signature), avec OCSP présentant un coût presque nul. La méthode 2-3 CRT présente un coût en calcul légèrement supérieur, car l'AC effectue en plus des opérations de hachage. Les

variantes de CRL, en particulier Over-issued CRL et CRL DP, ont un coût de calcul élevé car elles nécessitent davantage d'opérations de signature.

Les méthodes OCSP, CRL et SEM présentent le coût en calcul au niveau de l'EAGC (serveurs de publication, SEM et OCSP) le plus bas (vérification d'une signature), avec pour OCSP un coût pratiquement nul. La méthode 2-3 CRT est plus coûteuse car elles nécessite en plus de la vérification de signature, des opérations de hachage.

Coût en requête de vérification (CR)

Ce sont les méthodes SEM et OCSP qui consomment le moins de bande passante (une signature par requête). Vient ensuite la méthode 2-3 CRT (une branche d'arbre par requête) et, en dernier lieu, on retrouve la méthode CRL et ses dérivées qui consomment plus de bande passante.

Le coût en calcul au niveau des ACs est nul pour toutes les méthodes. De même, le coût en calcul au niveau des EAGCs est nul au niveau de toutes les méthodes, sauf pour OCSP et SEM pour lesquelles il faut compter le coût en signature du serveur OCSP et du serveur SEM (coût élevé).

L'approche SEM présente un coût nul en calcul au niveau des vérificateurs. Les méthodes CRL et OCSP ont un coût de vérification d'une signature. La méthode 2-3 est la plus gourmande car elle nécessite en plus du coût de vérification d'une signature, un certain nombre d'opérations de hachage.

Extensibilité (scalability)

Il est difficile de déterminer l'extensibilité des méthodes, car une approche peut être extensible lors d'une mise à jour d'informations de révocation et ne pas l'être lors d'une requête de révocation ; une approche peut être extensible par rapport à la faible bande passante consommée, mais son extensibilité peut être sérieusement remise en question à cause d'un important coût en calcul au niveau des ACs, EAGCs ou vérificateurs. Dans cette partie, nous étudions l'extensibilité des approches d'un point de vue global.

La CRL souffre beaucoup du problème d'extensibilité car sa taille augmente avec le nombre de certificats gérés, par conséquent elle est la moins extensible des méthodes. Pour y remédier, plusieurs variantes de la CRL : Δ-CRL, CRL DP, Over-issued CRL ont été proposées. La Δ-CRL, la CRL DP améliorent partiellement l'extensibilité de la CRL en réduisant la bande passante nécessaire à une requête où lors d'une mise à jour. La méthode Over-issued CRL minimise le risque d'implosion de requêtes CRL (cf. section 3.1.3).

Les méthodes SEM et OCSP ne sont pas a priori extensibles du fait du fort niveau de sollicitation des serveurs OCSP et SEM qui doivent signer toutes leurs réponses, une signature ayant un coût en temps de calcul important.

La méthode 2-3 CRT apparaît comme une méthodes ayant une bonne extensibilité avec un bon équilibre entre les différents coûts. Toutefois, 2-3 CRT est un peu plus gourmande en calcul au niveau des vérificateurs que les autres méthodes.

Risque système

A l'exception de OCSP et SEM qui ont recours à la mise en ligne de serveurs sensibles, les autres méthodes de révocation présentent un faible risque système car aucune TPC en ligne n'est utilisée.

Déni de Service (DoS)

Toutes les méthodes souffrent du problème de déni de service. Dans le cas de la CRL et de ses variantes, un utilisateur malicieux peut inonder un répertoire de publication avec des requêtes de CRLs, ce qui monopolise toutes les ressources réseaux, et prive d'autres utilisateurs de l'accès au répertoire. Ces utilisateurs légitimes se voient ainsi dans l'obligation de refuser des certificats, car ils ne peuvent plus les vérifier.

Pour la méthode 2-3 CRT, en cas d'inondation opérée sur le répertoire de publication, un utilisateur se trouve aussi dans l'incapacité de récupérer les branches de l'arbre CRT correspondant à l'état du certificat à vérifier. Noter que pour arriver au même niveau de saturation que pour les CRLs, l'attaquant devra envoyer ces requêtes à un rythme beaucoup plus soutenu car les réponses attendues sont moins volumineuses.

Dans le cas de la méthode OCSP, le serveur OCSP voit sa capacité de calculs (temps CPU) monopolisée à signer des réponses à des requêtes malicieuses (N'oublions pas qu'une opération de signature est très gourmande en temps de calcul). Pour la méthode SEM, le même phénomène se produit du fait que le serveur SEM doit contribuer à chaque opération de signature ou de déchiffrement demandée par un utilisateur sans vérification de la provenance de la demande.

Comme on peut le constater, les attaques de type déni de service peuvent avoir un impact sur la bande passante réseau (ressources réseaux des répertoires de publication), ou sur les ressources de calculs (temps CPU) des serveurs de confiance (OCSP, SEM) qui sont ainsi rendus (momentanément) inaccessibles aux utilisateurs légitimes.

Pour éviter certains dénis de service, il faudrait mettre en place un premier niveau de filtrage qui permettrait de distinguer les requêtes légitimes des requêtes malicieuses. Noter que pour éviter que ce premier filtre ne participe lui-même à une dégradation du service en cas d'inondations, il est nécessaire que cette première opération soit la plus légère possible en calculs ; elle devrait permettre d'éliminer la plupart des attaques n'ayant pas nécessité de gros moyens ; pour les attaques plus sérieuses, il est alors utile d'effectuer un second niveau de filtrage avec authentification de l'origine de la demande basée sur une signature par exemple ; en cas d'abus de la part d'un utilisateur, il serait alors possible de lui révoquer son certificat. Noter cependant que cette solution reste limitée puisque, dans leur principe, certains serveurs (serveurs de publication de CRLs, serveurs OCSP) doivent rester totalement accessibles.

Schéma On-line/Off-line

Certaines méthodes comme la CRL, et ses dérivées satisfont un schéma Off-line, tandis que les méthodes OCSP, SEM et CRL DP ne peuvent s'adapter qu'à un schéma On-line, avec un degré moindre pour CRL DP. La méthode 2-3 CRT peut être considérée comme une méthode hybride et peut donc avoir une double utilisation.

Le tableau 3 récapitule et compare les principales caractéristiques des méthodes de révocation.

#XXX Dans le texte ici ou sur la figure, expliquer la signification des +, ++, ++++

Que signifie N et O ? A préciser aussi dans l'article

	TPC	**Extensibilité**	**Bande Passante consommée**		**Coût en calcul**			**Complexité**	**1/2**
			AC/EAGC	EAGC/U	AC	EAGC	U		
CRL	N	+	++++	++++	+	+	+	+	2
Delta-CRL	N	++	++++	++	++	+	+	+	1 ; 2

CRL DP	N	++	++++	+++	++	+	+	+	1
Over-issued CRL	N	++	++++	++++	++	+	+	+	2
2-3 CRT	N	++++	+	++	++	++	++	+	1 ; 2
SEM	O	+	++++	++	+	+++	+	++	1
OCSP	O	++	ε	+	+	+++	+	+	1

TPC : Tierce Partie de Confiance
AC : Autorité de Certification
EAGC : Répertoire de publication, serveur OCSP ou serveur SEM
ε : non considéré
U : Utilisateur
1 ; 2 : 1 schéma On-line ; 2 : schéma Off-line

Tableau.3 Tableau récapitulatif/comparatif des méthodes de révocation

7 Conclusions

L'utilisation des certificats électroniques fait de plus en plus partie intégrante de notre vie quotidienne. Toutefois, cette intégration ne sera pleinement réussie que si une condition primordiale est satisfaite : garantir l'état de révocation des certificats.

Cet article décrit un ensemble représentatif de méthodes de révocation et compare ces méthodes en fonction de différents critères d'évaluation comme le coût en calcul, le coût en bande passante, la fraîcheur de révocation, l'extensibilité de l'approche, etc.

Le choix d'une méthode de révocation par rapport à une autre doit être fait en fonction des exigences de l'application (surtout en matière de fraîcheur d'informations de révocation et niveau de sécurité), le niveau de disponibilité des ressources dans le système (bande passante, puissance de calculs, etc.), le délai de calcul toléré, ainsi que d'autres facteurs. Dans ce rapport, nous donnons les moyens nécessaires aux ingénieurs, concepteurs et décideurs pour réaliser ce choix, afin de satisfaire au mieux à leurs exigences environnementales.

Du fait du nombre limité de pages de cet article, nous n'avons pas pu traiter d'autres types de certificats tels que les certificats emboîtés, les certificats d'attributs et les certificats actifs. Pour une analyse plus complète, le lecteur intéressé pourra se reporter au rapport de recherche [Bek04].

BIBLIOGRAPHIE

[Arn00A] A. Arnes. *« Public Key Certificate Revocation Schemes »*. Phd thesis, Department of Telematics, Norwegian University of Science and Technology, February 2000.

[Arn00B] A. Arnes, M. Just, S.V. Knapskog, S. Lioyd, H. Meijer. *« Selecting Revocations Solutions for PKI »*. Norwegian University of Science and Technology, March 2000.

[Bek04] C. Bekara, M. Laurent-Maknavicius, *« Méthodes de révocation des certificats numériques »*, Rapport de recherche 04015 LOR, 2004.

[Bon01] D. Boneh, X.Ding, G. Tsudik, C. Ming Wong. « *A Method for Fast Revocation of Public Key Certificates and Security Capabilities*». In Proceedings of the 10th Usenix Security Symposium, pp. 297-308, Washington DC, 2001.

[Kik99] H. Kikuchi, K. Abe, S. Nakanishi. « *Performance Evaluation of Public-key Certificate Revocation System with Balanced Hash Tree* ». International Workshops on Parallel Processing, Wakamatsu, Japan, September 1999.

[Mun04] J. L. Munoz, J. Forne, O. Espazara, M. Soriano. "Certificate revocation system implementation based on the Merkle hash tree". International Journal of Information Security. Volume2, Number2, January 2004, ISSN/ISBN1615-5262.

[Nao00] M. Naor, K. Nissim. « *Certificate Revocation and Certificate Update* ». IEEE Journal on Selected Areas in Communications, Vol. 18, No. 4, 2000.

[rfc2560] RFC 2560 – " X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP ". http://www.faqs.org/ftp/rfc/pdf/rfc2560.txt.pdf, June 1999.

[Zhe03] P. Zheng. « *Tradeoffs in Certificate Revocation Schemes* ». ACM SIGCOMM CCR, Vol. 33, No. 2, pp. 103-112, New York, 2003.