Protocoles d`Authentification

Refik Molva et Yves Roudier

Institut EURECOM, BP 193 Sophia Antipolis Cedex - France

*{*refik.molva@eurecom.fr , yves.roudier@eurecom.fr*}*

Résumé : cet article décrit les techniques d'authentification qui permettent d'obtenir une assurance sur l'identité d'un interlocuteur. Trois grandes classes de techniques actuellement employées sont décrites : les techniques d'authentification faibles utilisant les mots de passe, les techniques d'authentification cryptographique basées sur les protocoles question-réponse et les techniques d'authentification à base de dispositifs matériels.

Introduction

Cet article présente les techniques par lesquelles une entité peut obtenir une assurance sur l'identité de son interlocuteur dans un environnement potentiellement hostile où une entité malveillante peut tenter d'usurper cette identité. Ces techniques sont fréquemment appelées identification, authentification d'entité ou vérification d'identité. L'authentification d'entité ne vise pas à protéger un message particulier ni à fournir une preuve irrévocable sur un laps de temps conséquent comme dans le cas de l'authentification de message. L'authentification d'entité, qui sera seule traitée dans cet article où elle sera simplement appelée authentification, corrobore l'identité d'un correspondant en temps réel.

Tout schéma d'authentification suppose au moins deux parties : un demandeur, qui présente une identité, et un vérificateur, qui s'assure de sa validité. Un schéma d'authentification doit permettre la validation de l'identité d'une entité légitime en présence d'attaques possibles comme par exemple le déguisement, c'est-à-dire le rejeu par un attaquant des messages émis par l'entité légitime afin d'usurper l'identité de cette dernière. Afin de contrer ces attaques, le schéma d'authentification doit fournir des garanties de sécurité qui permettent au vérificateur de corroborer l'identité affichée par un demandeur. Ces garanties peuvent reposer sur : 1) la localisation du demandeur, 2) ce que sait le demandeur, 3) ce que possède le demandeur, ou 4) ce qu'est le demandeur. Certaines garanties peuvent être prouvables. La classification des techniques d'authentification permet de dégager quatre grandes catégories basées sur la nature de la sécurité qu'elles mettent en œuvre :

- techniques d'authentification faible exploitant des informations de taille limitée et/ou non aléatoires ;
- techniques d'authentification forte basées sur des méthodes cryptographiques ;
- techniques d'authentification forte basées sur des dispositifs matériels ;
- techniques d'authentification biométrique, directement liées aux caractéristiques physiologiques ou aux traits comportementaux d'un individu

Authentification faible

Les mots de passe constituent une technique d'authentification unidirectionnelle très répandue : un utilisateur fournit son identité et un mot de passe pour accéder à une ressource. Un mot de passe constitue donc un secret partagé entre l'utilisateur et le système auprès duquel il s'authentifie : prouver qu'il connaît ce secret donne l'assurance que son identité est correcte. La faiblesse principale de cette technique provient justement de ce que les mots de passe peuvent facilement être dévoilés ou découverts. Les systèmes d'authentification par mot de passe sont sujet à plusieurs types d'attaques, en particulier la recherche exhaustive de mots de passe à partir de leur texte chiffré et le rejeu de mots de passe.

Le stockage des mots de passe sous forme de fichiers chiffrés constitue une précaution élémentaire mais autorise cependant les attaques exhaustives. Il est par exemple possible d'essayer exhaustivement tous les mots de passe possibles et pour chacun, de comparer son chiffrement à la valeur du chiffrement du mot de passe de l'utilisateur stockée dans le fichier. Le salage de mots de passe [MT 79], par exemple utilisé dans Unix, améliore la sécurité de ce schéma : il consiste à ajouter à chaque mot de passe, lorsqu'il est introduit initialement, une chaîne de t bits aléatoires, appelée "sel", avant d'appliquer une fonction de hachage à sens unique. Le mot de passe haché et le "sel" sont enregistrés dans le fichier de mots de passe. Le salage d'un mot de passe n'augmente pas la difficulté de recherche par une attaque exhaustive sur un seul mot de passe puisque le "sel" est conservé en clair

dans le fichier de mots de passe ; par contre, elle élève le coût d'une attaque simultanée sur plusieurs mots de passe, puisqu'il faut prévoir 2^t variations de chaque mot de passe.

Pour éviter les problèmes de rejeu, c'est-à-dire l'interception et la retransmission d'un mot de passe par un intrus cherchant à usurper l'identité d'une entité légitime, les mots de passe fixes ne doivent être utilisés que sur des transmissions sécurisées. Les mots de passe variables constituent une alternative intéressante aux mots de passe fixe vers un schéma question-réponse plus apte à résoudre les problèmes de rejeu. Dans un système à mots de passe variables, chaque mot de passe échangé en texte clair n'est valable qu'à une seule et unique utilisation : ce type de technique met l'utilisateur à l'abri des adversaires passifs effectuant seulement des écoutes et utilisant les mots de passe interceptés pour effectuer une attaque par rejeu. Par contre, un adversaire actif qui intercepte et bloque les communications et envoie ses propres données à la place peut tout de même déjouer ce type d'authentification. On peut cataloguer les schémas de mots de passe variables selon trois modes de fonctionnement :

- l'utilisateur reçoit du système une liste de mots de passe secrets à usage unique
- un mot de passe (secret) est partagé au départ par le système et l'utilisateur ; à chaque authentification, l'utilisateur envoie au système un nouveau mot de passe chiffré par une clé dérivée du mot de passe précédemment partagé.
- l'utilisateur et le système partagent le résultat secret du hachage par une fonction à sens unique, itéré un certain nombre de fois, d'un mot de passe initial connu de l'utilisateur seul.

L'algorithme de Lamport [LAM 81] est un exemple de ce dernier type de système de mots de passe variables. Il nécessite une phase d'initialisation préalable avant d'être employé pour l'authentification proprement dite. Lors de l'initialisation, l'utilisateur choisit un mot de passe m secret. Une constante t fixe le nombre d'authentification permises avec m. L'utilisateur calcule le hachage de m par une fonction à sens unique h, itéré t fois, et communique d'une manière qui en garantisse l'authenticité $h^t(m)$ au système, qui stocke ce message. Pour sa i-ème authentification après l'initialisation (voir Figure 1), l'utilisateur transmet au système auprès duquel il cherche à s'identifier un message portant son identité, i, et $h^{t-i}(m)$. Le système vérifie que i correspond bien à la i-ème tentative d'authentification (c'est-à-dire à la i-ème variation du mot de passe initial) pour l'utilisateur en question et que $h(h^{t-i}(m_{reçu}))$ est bien égal à $h^{t-i+1}(m)$ dont la valeur est connue par le système. Si la vérification est réussie, le système incrémente le nombre de tentatives réussies et conserve la valeur $h^{t-i}(m)$ reçue lors de cette authentification. Lorsque l'utilisateur arrive au nombre maximum t d'authentifications possibles, il doit réinitialiser le mécanisme avec un nouveau mot de passe afin d'éviter le rejeu possible.

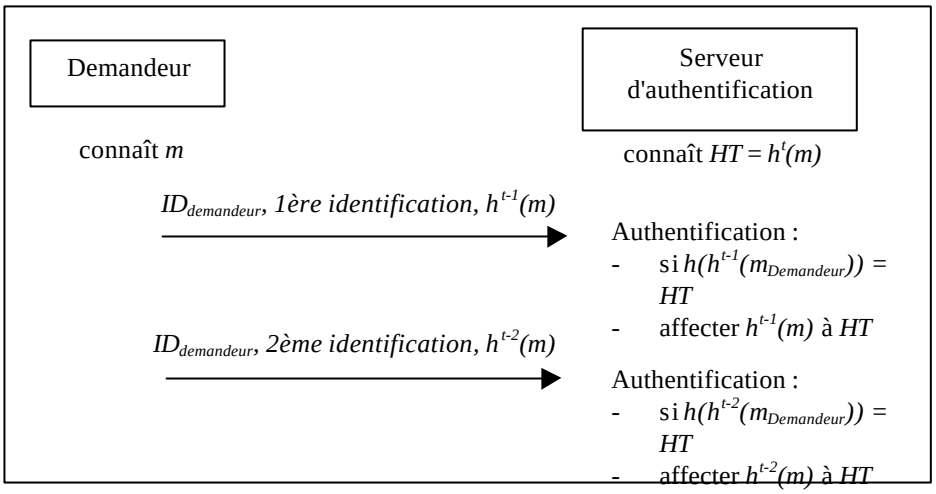

Figure 1. Authentification par l'algorithme de Lamport

Authentification forte

Lorsqu'une entité effectue une authentification forte auprès d'une autre entité, elle prouve qu'elle connaît un secret associé avec son identité déclarée. Cette preuve s'appuie sur l'utilisation de techniques telles que les fonctions de hachage, ou la cryptographie symétrique ou asymétrique : on parle de protocole cryptographique

question-réponse. Un protocole question-réponse fonctionne d'après le principe suivant : l'entité B qui joue le rôle de vérificateur choisit de manière aléatoire une donnée, appelée **question**, qui est envoyée à l'entité A qui doit prouver son identité ; l'entité A applique à son tour à la question une opération cryptographique basée sur un secret qu'elle détient ; le résultat de cette opération, appelé **réponse**, est renvoyé à B pour fournir la preuve de l'identité de A. Le but principal des protocoles question-réponse est d'empêcher une famille d'attaques connue sous le nom de **rejeu**. Le rejeu décrit la retransmission par un intrus, dans le but d'usurper l'identité du demandeur, d'une réponse qui a déjà été utilisée entre deux entités légitimes comme réponse à une nouvelle question . Afin d'assurer la protection contre le rejeu, la réponse calculée par le demandeur A doit être différente à chaque exécution du protocole d'authentification. La technique permettant d'obtenir un paramètre qui varie dans le temps constitue un des éléments essentiels d'un protocole d'authentification. Le second élément important d'un protocole d'authentification concerne la manière dont le demandeur A calcule la réponse correspondant à la question posée par le vérificateur B. Plusieurs variantes de calcul existent en fonction de la méthode cryptographique utilisée. Nous présentons les protocoles correspondant aux cas les plus significatifs en utilisant la notation suivante :

X : représentation binaire de l'identité de l'entité X

t_X : cachet d'horodatage émis par l'entité X

s_X : numéro de séquence émis par l'entité X

n_X : nombre aléatoire émis par l'entité X

$E_K(m_1, m_2, ... m_n)$: chiffrement par un algorithme symétrique de la chaîne binaire constituée par la concaténation des messages $m_1, m_2, ... m_n$

$h_K(m_1, m_2, ... m_n)$: résultat du hachage avec la clé K de la chaîne binaire constituée par la concaténation des messages $m_1, m_2, ... m_n$

K_{XY} : clé secrète partagée par les entités X et Y

Protocoles avec secret partagé

Dans le protocole ci-dessous utilisant l'horodatage, l'entité B retrouve la valeur de t_A en déchiffrant le message :

$A \rightarrow B : A, E_{Kab}(t_A , B)$

avec la clé Kab qui identifie l'entité A auprès de B. A est authentifié si t_A appartient à l'intervalle d'acceptation courant chez B. L'inclusion de l'identité de B dans le message d'authentification est une technique très simple qui permet d'éliminer les attaques qui consisteraient à effectuer un rejeu de ce message dans une session parallèle établie vers A par un intrus qui voudrait passer pour B. La version suivante du protocole d'horodatage utilise une fonction de hachage à la place du chiffrement :

$A \rightarrow B : A, h_{Kab}(t_A , B), t_A$

Dans ce cas, l'envoi de t_A en clair est nécessaire afin de permettre à B de reconstituer tous les paramètres d'entrée de la fonction de hachage, la fonction de hachage n'étant pas inversible.

Une autre variante de ce protocole est obtenue en utilisant un numéro de séquence comme paramètre variable dans le temps à la place du cachet d'horodatage.

$A \rightarrow B : A, E_{Kab}(s_A , B)$ ou $h_{Kab}(s_A , B)$

La génération de ce paramètre par A et sa vérification par B sont effectuées en utilisant des compteurs locaux qui sont synchronisés entre eux. La vérification du message se fait comme pour l'horodatage avec la différence suivante : la valeur de s_A est retrouvé par B en utilisant le compteur local.

Le protocole à base de nombre aléatoire s'affranchit de la nécessité de maintenir des horloges ou des compteurs synchronisés par l'introduction d'une question explicite envoyée par le vérificateur. Par conséquent deux messages sont nécessaires à sa mise en œuvre.

1. $B \rightarrow A : n_B$

2. $A \rightarrow B : A, E_{Kab}(n_B , B)$ ou $h_{Kab}(n_B , B)$

La vérification de la réponse se fait en utilisant le nombre aléatoire contenu dans l'état temporaire mémorisé par B. Le désavantage d'une interaction explicite avec deux messages qui est propre à la technique des nombres aléatoires est réduit dans le cas de l'authentification mutuelle où une interaction minimum avec deux messages est nécessaire par définition :

1. $A \rightarrow B : A, n_A$

2. $B \rightarrow A : B, n_B, E_{Kab}(n_A , A)$ ou $h_{Kab}(n_A , A)$

3. $A \rightarrow B : E_{Kab}(n_A, n_B , B)$ ou $h_{Kab}(n_A, n_B , B)$

Il faudrait noter dans le troisième message de ce protocole la présence des deux nombres aléatoires n_A et n_B qui est nécessaire pour éviter des attaques basées sur une exécution parallèle du même protocole afin d'obtenir les réponses en utilisant les entités légitimes comme un oracle. Si le troisième message de ce protocole avait le même format que le second message (si n_A n'y était pas pris en compte pour le calcul du résultat), l'attaquant X pourrait mettre en œuvre le scénario suivant pour passer pour A auprès de B :

1. (session 1) $X \rightarrow B : A, n_X$
2. (session 1) $B \rightarrow X : B, n_B, E_{Kab}(n_X , A)$ ou $h_{Kab}(n_X , A)$
3. (session 2) $X \rightarrow A : B, n_B$
4. (session 2) $A \rightarrow X : A, n_A, E_{Kab}(n_B , B)$ ou $h_{Kab}(n_B , B)$
5. (session 1) $X \rightarrow B : E_{Kab}(n_B , B)$ ou $h_{Kab}(n_B , B)$

Dans ce scénario, la deuxième session, établie par l'attaquant en passant pour B auprès de l'entité A, permet à l'attaquant d'obtenir la réponse à la question (n_B) posée par B dans la première session en utilisant A comme oracle. Cette attaque réussit parce que le second message de la deuxième session peut parfaitement être utilisé en tant que troisième message de la première session. Malgré leur simplicité apparente, la conception des protocoles d'authentification résistant à des attaques similaires est une tâche assez complexe. [BGH 93] présente une analyse étendue des attaques correspondantes et une méthode de conception permettant de les éviter.

Protocoles à base de clés publiques

Deux méthodes se trouvent à la base des protocoles d'authentification utilisant les algorithmes à clés publiques :

- le demandeur chiffre (ou signe) la question avec sa clé privée et la réponse résultante est déchiffrée par le vérificateur en utilisant la clé publique du demandeur ;
- le demandeur déchiffre avec sa clé privée une question qui a été chiffrée par le vérificateur en utilisant la clé publique du demandeur.

La deuxième méthode qui requiert l'échange d'une question explicite est bien appropriée pour la technique des nombres aléatoires. Les trois techniques de génération de paramètre variable dans le temps peuvent être mises en œuvre par la première méthode. Les protocoles d'authentification à base de clés publiques sont de plus en plus répandus dans la mise en œuvre des nouveaux protocoles de communication en raison de l'absence de besoin de distribution de secret partagé et grâce au développement des infrastructures de certification qui permettent une utilisation sécurisée des clés publiques.

Protocoles utilisant un serveur d'authentification

Le concept de serveur d'authentification résoud principalement le problème de la distribution du secret partagé dans le cas des protocoles à base de secret partagé. Le premier protocole d'authentification utilisant un serveur a été introduit par Needham et Schroeder [NS 78]. Dans ce protocole, au lieu de partager un secret différent avec chacun de ses correspondants potentiels, chaque entité partage seulement un seul secret avec le serveur d'authentification appelé S. Quand une entité (A) a besoin de s'authentifier auprès d'une autre (B), A contacte S comme suit :

1. $A \rightarrow S : A, B, n_A$
2. $S \rightarrow A : E_{Ka}(n_A , B, Kab, ticket= E_{Kb}(A, Kab))$
3. $A \rightarrow B : ticket= E_{Kb}(A, Kab)$
4. $B \rightarrow A : E_{Kab}(n_B)$
5. $A \rightarrow B : E_{Kab}(n_B -1)$

Un nouveau secret Kab qui est valable pour la durée de cette instance du protocole est généré par le serveur et envoyé à A dans le message 2. qui est chiffré avec le secret individuel Ka partagé par A et S. Ce message contient également un autre champ chiffré appelé **ticket** qui est une enveloppe chiffrée par la clé Kb de B et contenant la nouvelle clé Kab. A la réception du message 2., A découvre la valeur de Kab par le déchiffrement du

message en utilisant la clé Ka. Par ce déchiffrement, A obtient aussi la valeur du ticket qu'elle transmet directement à B. D'une manière similaire, B retrouve la nouvelle clé partagée Kab en déchiffrant le ticket avec sa clé individuelle Kb. Les messages 4. et 5. constituent un protocole pour l'authentification unidirectionnelle de A par B utilisant le secret partagé Kab.

Le protocole de Needham-Schroeder présente cependant une faiblesse : aucun élément de ce protocole ne permet de garantir que la clé Kab est nouvelle. Ainsi, dans le cas où la valeur de la clé Kab correspondant à une exécution antérieure du protocole serait dévoilée par mégarde à un attaquant, celui-ci peut parfaitement réussir à passer pour A auprès de B en effectuant un rejeu du message 3. et en exécutant le reste du protocole en utilisant la valeur de Kab. Cette faiblesse a été remarquée par les inventeurs de ce protocole [NS 87] et une version corrigée du protocole incluant quelques améliorations a été mise en œuvre par le système Kerberos [KN 93]. L'amélioration principal du protocole de Kerberos consiste à inclure un cachet d'horodatage dans le ticket afin de limiter la durée de vie de la clé Kab. Plusieurs variantes de protocoles d'authentification avec un serveur ont été proposées [OR 87] [BGH 95].

Dispositifs personnels

L'authentification forte repose sur la connaissance par le demandeur d'un secret qui résiste à diverses attaques concernant la méthode cryptographique utilisée par le protocole d'authentification. Dans le cas des techniques symétriques, un secret de 90 bits aléatoirement choisi est considéré comme sûr à l'état actuel de la technologie. D'une façon similaire, pour un algorithme asymétrique, une clé privée sûre doit au minimum contenir 768 bits. L'exigence concernant la taille du secret pose un problème dans le cas où l'entité qui exécute le protocole d'authentification forte est un système public qui agit au nom d'un utilisateur humain (terminal public, distributeur de billets, etc.) :

- d'une part, le secret ne peut pas être mémorisé par l'utilisateur humain puisque contrairement aux informations redondantes qui sont faciles à mémoriser pour l'homme, un secret sûr est une donnée aléatoire ou pseudo-aléatoire qui ne présente pas de redondance,
- d'autre part, un secret ne peut être stocké dans la mémoire d'un système public pour des raisons évidentes.

La solution à ce problème consiste à doter chaque utilisateur d'un dispositif personnel qui lui permet de mettre en œuvre un protocole d'authentification forte à base de secrets sûrs sans toutefois nécessiter leur mémorisation par l'utilisateur. Nous considérons deux types de dispositifs qui peuvent répondre à ce besoin :

- les dipositifs passifs ont la seule fonction de mémoire et
- les dipositifs actifs qui possèdent les fonctions de mémoire et de calcul qui leur permettent d'agir comme une entité d'authentification active.

Les dispositifs passifs contiennent une portion de mémoire sur laquelle sont inscrites les données nécessaires à la vérification de l'identité d'un utilisateur. En utilisant ces données, plusieurs systèmes d'authentification publics peuvent procéder à l'authentification forte de l'utilisateur de la façon suivante : le système d'authentification vérifie le secret fourni par l'utilisateur (PIN) par rapport à l'identité (ID) fournie par le dispositif en utilisant une clé secrète K qui est commune à tous les systèmes d'authentification. Le PIN est un secret faible et facile à mémoriser par un utilisateur humain tandis que K est un secret sûr dont la découverte par recherche exhaustive est impossible. La valeur d'ID et le résultat d'une fonction à sens unique (FSU) calculée sur ID, PIN et K sont stockés sur le dispositif et peuvent être lus par un simple lecteur de dispositif. L'expression h(PIN, ID, K), utilisant une fonction de hachage comme MD5, peut être un exemple pour une FSU. Le caractère "à sens unique" de la FSU assure que les paramètres d'entrée ne peuvent être retrouvés à partir du résultat de la fonction.

Grâce à la fonction à sens unique, la possession du dispositif sans la connaissance du PIN ne permet pas à un intrus de s'authentifier en passant pour l'utilisateur légitime. Le vol du dispositif ne permet de découvrir la valeur du PIN par une recherche exhaustive de toutes les valeurs, même si le PIN constitue un secret faible, puisque la connaissance de l'autre secret K, qui est un secret sûr, est aussi nécessaire pour tester les valeurs candidates pour le PIN. Le test exhaustif qui consiste simplement à rentrer des valeurs de PIN à titre d'essai dans un système d'authentification n'est pas permis, le nombre d'essais étant limité à un petit nombre sur chaque système. Dans ce processus de vérification, le dispositif personnel joue principalement le rôle d'une mémoire déportée pour le système d’authentification et permet une gestion distribuée de la population d'utilisateurs. Le désavantage principal de ce schéma est le partage du secret K par tous les systèmes d'authentification qui constitue un point

de faiblesse commun pour l'ensemble. L'exemple le plus connu de tels dispositifs est le système de cartes bancaires à bande magnétique.

Les dispositifs actifs possèdent, en plus de la capacité de mémorisation, une capacité de calcul qui leur permet d'agir comme une entité indépendante dans le processus d'authentification. Les secrets mémorisés par un dispositif actif ne sont pas accessibles en lecture pour une entité externe ; seul le dispositif lui-même peut accéder aux secrets contenus dans sa mémoire. Ainsi le dispositif actif communique avec les entités externes en échangeant des messages d'authentification définis d'après un des protocoles présentés précédemment dans cet article. Conformément au principe d'authentification forte, un dispositif actif ne divulgue jamais le secret mais en démontre sa connaissance en répondant à des questions.

L'authentification forte d'un utilisateur par une entité distante et à travers un dispositif actif s'effectue en deux phases :

- dans une première phase, le dispositif actif procède à la vérification de l'utilisateur sur la base du secret faible (PIN) et à travers un médium de communication protégé comme dans le cas d'un dispositif passif ;
- une fois l'authentification de l'utilisateur par le dispositif réalisée avec succès, le dispositif agit au nom de l'utilisateur en mettant en œuvre un protocole d'authentification forte à base d'un secret sûr (K) mémorisé dans le dispositif pour s'authentifier auprès d'autres entités distantes à travers un médium de communication qui est éventuellement exposé à des attaques.

La première phase de ce processus présente un avantage par rapport à l'authentification avec un dispositif passif : il n'est pas nécessaire de placer un secret commun dans plusieurs systèmes d'authentification. En effet, grâce à sa capacité de calcul, le dispositif actif peut assurer la vérification de l'utilisateur et une garantie contre la recherche exhaustive sans que la lecture par un système externe des secrets contenus dans le dispositif soit nécessaire. La mise en œuvre la plus courante du principe de dispositif actif se trouve dans les cartes à puce à microprocesseur.

Conclusion

Les techniques décrites dans cet article sont aujourd'hui devenues classiques. Le choix de l'une d'entre elles dépend des ressources à protéger et de la facilité de mise en œuvre et d'emploi pour l'utilisateur. De nouvelles techniques, biométriques, c'est-à-dire cherchant à identifier des caractéristiques physiques ou comportementales des individus, commencent à se répandre et complètent utilement la panoplie des techniques d'authentification. La biométrie n'étant pas étudiée dans cet article, le lecteur peut se référer à [JHK 00] pour plus d'informations.

Bibliographie

[BGH 93] R. BIRD, I. GOPAL, A. HERZBERG, P. JANSON, S. KUTTEN, R. MOLVA, M. YUNG, *Systematic Design of a Family of Attack-Resistant Authentication Protocols*, IEEE Journal on Selected Areas in Communications, special issue on Secure Communications, vol. 11 numéro 5, pages 679-693, Juin 1993

[BGH 95] R. BIRD, I. GOPAL, A. HERZBERG, P. JANSON, S. KUTTEN, R. MOLVA, M. YUNG, *The KryptoKnight family of lightweight protocols for authentication and key distribution*, IEEE/ACM Transactions on Networking, vol. 3, numéro 1, pages 31-41, Février 1995

[JHK 00] JAIN A. K., HONG L., PANKANTI S., *Biometrics: Promising Frontiers for Emerging Identification Market*, Communications of the ACM , pages 91-98, février 2000

[KN 93] KOHL J., NEUMANN C., RFC 1510, *The Kerberos Network Authentication Service (V5)*, Internet Request for Comments 1510, Septembre 1993

[LAM 81] LAMPORT L., *Password Authentication with Insecure Communications*, Communications of the ACM, numéro 24, pages 770-772, 1981

[MT 79] MORRIS R., TOMPSON K., *Password Security : A Case History*, Communications of the ACM , numéro 22, pages 594-597, 1979

[NS 78] R. M. NEEDHAM, M.D. SCHROEDER, *Using encryption for authentication in large networks of computers*, Communications of the ACM , numéro 21, pages 993-999, 1978

[NS 87] R. M. NEEDHAM, M.D. SCHROEDER, *Authentication revisited*, Operating Systems Review, numéro 21, 1987

[STI 96] DOUGLAS STINSON, *Cryptographie Théorie et Pratique*, International Thomson Publishing France, 1996

[OR 87] D. OTWAY, O. REES, *Efficient and timely mutual authentication*, Operating Systems review, vol. 21, numéro 1, pages 8-10, Janvier 1987