MajecSTIC 2009
Avignon, France, du 16 au 18 novembre 2009

Authentification HTTP Digest SIP renforcée

Thomas Guillet[1] et Ahmed Serhrouchni[1]

1 : TELECOM ParisTech, Département Informatique et Réseaux, 46 rue Barrault, 75634 Paris Cedex 13 - France.

Contact : guillet@telecom-paristech.fr

Résumé
Le réseau Internet devient d'une manière certaine le réseau de transport de tout type de média. La téléphonie suit naturellement cette tendance, ce service migre donc progressivement vers ce réseau. Mais comme le « Web » ou la messagerie, la téléphonie sur Internet n'échappe pas aux problèmes de sécurité. Plusieurs travaux ont établi cette problématique à laquelle s'ajoute celle des failles traditionnelles des réseaux ouverts basés sur le protocole IP. L'enjeu est donc de préserver la confiance des usagers pour un service aussi emblématique que le téléphone.

Le protocole qui semble le plus émerger est le standard de l'IETF SIP. Ce dernier spécifie la signalisation pour l'établissement d'un appel et les modalités pour le transport de la voix. Une large communauté a contribué à sécuriser l'environnement SIP, principalement en rajoutant de nouveaux paramètres ou en préconisant l'utilisation de protocoles de sécurité pour le transport des messages SIP ou de la voix. Ces propositions ont un coût en temps de calcul, en bande passante et ne sont pas toujours interopérables avec les implémentations existantes. Notre étude a donc recherché les opportunités de renforcer la sécurité, en particulier l'authentification, sans modifier les échanges SIP et en garantissant une totale interopérabilité avec les infrastructures existantes. Ce cahier des charges nous a permis de proposer une solution validée formellement par l'outil AVISPA et sur une plate-forme logicielle basée sur le logiciel libre ASTERISK.

Abstract
The Internet network becomes certainly the transport network of all types of media. Telephony is no exception to this trend, the service is gradually migrating to this network. As the "Web" or mail, Internet telephony is not immune to security problems. Several studies have established the problem, plus that of the traditional shortcomings of open networks based on IP. The challenge is to maintain the confidence of users for one as iconic as the telephone.

The most emerged protocol is the IETF standard SIP. This latter specifies the signaling for call establishment and arrangements for the transport of voice. A large community has helped to secure the SIP environment, mainly by adding new parameters or advocating the use of secure protocol for the transport of SIP messages or voice. These proposals have a cost in computation time, bandwidth and are not always interoperable with existing implementations. Our study therefore sought opportunities to enhance security, especially authentication, without modifying the SIP exchanges and ensuring full interoperability with existing infrastructures. This specification allowed us to propose a solution formally validated by the tool AVISPA and a software platform based on open source Asterisk.

Mots-clés : Téléphonie, IP, sécurité, SIP, authentification.

Keywords: Telephony, IP, security, SIP, authentication.

1. Introduction

La téléphonie sur IP (Telephony over IP, ToIP) est sans aucun doute l'application après le Web et la messagerie qui confirmera encore plus l'infrastructure IP (Internet Protocol) comme le standard du transport de tout type d'information ou de média. L'adoption d'une infrastructure unique pour le transport de tout type de données représente un avantage économique très significatif pour les opérateurs. Ainsi, la migration des services de téléphonie classique vers le tout IP semble être incontournable.

L'insécurité des réseaux Internet est par ailleurs perçue comme une faiblesse suffisamment pénalisante pour avoir de nombreux détracteurs et pose ainsi pour les fournisseurs et les usagers de sérieuses inquiétudes. La téléphonie classique offrait des garanties de sécurité qui étaient dues en grande partie à son infrastructure support dédiée et en particulier à sa disponibilité. La crédibilité de la téléphonie sur IP passe donc par une offre aussi fiable que celle du Réseau Téléphonique Commuté (RTC).

Les principaux problèmes rencontrés pour la téléphonie sur IP sont les écoutes illicites, le déni de service, l'usurpation d'identité, l'usurpation de droits, le détournement d'appel voire même le SPAM téléphonique [18]. Certes, ces attaques existaient déjà avec la téléphonie classique, mais l'usage d'un réseau IP les rend plus facilement réalisables (accès distant et large diffusion des outils réseaux) et moins coûteuses. Des solutions existent comme le chiffrement de la voix ou de la signalisation, l'utilisation de mécanisme d'authentification, l'utilisation d'outil réseau classique comme les firewalls pour limiter les problèmes de sécurité.

Les solutions de sécurité rencontrent cependant de nombreuses difficultés d'implémentation et de déploiement. Le chiffrement de la signalisation ou de la voix [13] nécessite un mécanisme d'échange de clés universel ou une infrastructure de gestion de certificats pour pouvoir être utilisé par tous les usagers ou les serveurs. Le chiffrement nécessite du temps de calcul et augmente la taille des paquets IP, ce qui n'est pas toujours conciliable avec une application temps réel comme pour le transport de la voix pour une communication. La multitude de protocoles comme SIP [3], H323 [6] ou encore Skype [19] ne facilite pas non plus l'adoption d'un standard de sécurité.

La fédération des solutions de sécurité viendra peut-être de l'adoption massive actuelle du standard SIP de l'IETF (Internet Engineering Task Force). SIP (Session Initiation Protocol) est un protocole de signalisation de la téléphonie qui s'appuie sur d'autres standards pour compléter ses services (cf. figure 1). Il permet d'initier, de modifier et de terminer des sessions multimédias et donc des appels téléphoniques. Les spécifications de SIP incluent un volet sécurité qui s'appuie largement sur les solutions usuelles de l'Internet comme HTTP Digest [5], S/MIME [10], IPSec [7] ou TLS [4]. Mais là aussi il existe de nombreuses limitations [14]. TLS implique l'utilisation du protocole de transport orienté connexion TCP [9], alors que la signalisation SIP est généralement transportée par le protocole sans connexion UDP [8]. TLS et IPsec n'offrent pas de solution bout-en-bout sauf si l'appel s'effectue dans un domaine ou entre plusieurs domaines qui adoptent une même politique de sécurité : néanmoins il est difficile de s'assurer de la politique de sécurité du domaine auquel appartient son correspondant. S/MIME peut permettre l'authentification bout-en-bout des usagers à condition évidemment que ces derniers puissent échanger leur certificat. S/MIME permet également d'assurer l'intégrité des messages : cette propriété ne peut cependant pas s'appliquer à l'ensemble du message SIP puisque les serveurs relais peuvent rajouter de nouveaux champs. Les limitations sont donc nombreuses.

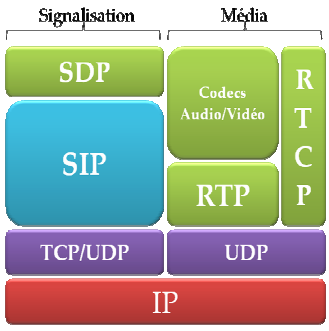


FIG. 1 – Pile protocolaire réseau dans un environnement SIP.

S'appuyant sur ce constat, notre contribution s'est donc focalisée sur les solutions envisageables pour sécuriser SIP tout en garantissant son déploiement. Nous exposerons dans un premier temps nos motivations et les choix effectués. Puis nous détaillerons une solution de renforcement validée formellement par le programme AVISPA [1] et implémentée sur une plate-forme logicielle. La conclusion évoquera enfin nos perspectives de recherches.

2. Motivations et choix pour renforcement de l'authentification SIP

La sécurité de la téléphonie sur IP et en particulier de SIP nous paraît fondamentale pour garantir la confiance des usagers et des opérateurs dans cette approche technique. Les gains économiques attendus par l'utilisation d'un seul réseau de transport ne peuvent suffire à eux seuls pour imposer un ralliement inconditionnel à la ToIP. L'existence de marché noir des écoutes téléphonique en Grèce et en Italie [16] montre que les problèmes de sécurité sont déjà une réalité et une nouvelle opportunité pour voir se développer la cybercriminalité.
L'architecture SIP (cf. figure2) se compose de clients (user agent), qui initient et reçoivent les appels, et de serveurs qui enregistrent ou localisent les clients (Registrar server, Location server), commutent ou redirigent les appels (Proxy server, Redirect server). L'établissement d'une communication se fait par l'échange de messages (requête ou réponse) entre les différents éléments du réseau. Une fois la session établie, l'échange des données s'effectuera quant à lui directement entre les deux clients SIP. Les échanges d'informations entre les différents acteurs d'une architecture SIP se font donc par des messages codés en utilisant la syntaxe de message HTTP/1.1 [12] et un codage UTF-8 [17] (cf. figure . Le flux voix est en général transporté selon les spécifications de RTP [15].

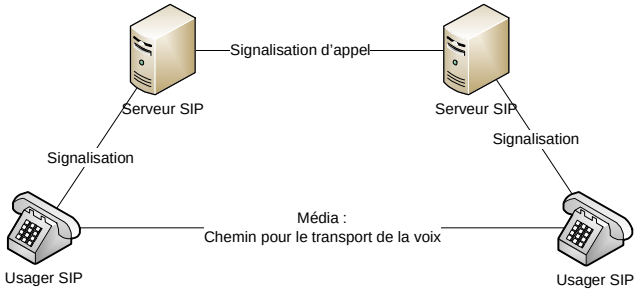


FIG. 2 - Architecture élémentaire SIP (trapézoïde).

Les solutions pour sécuriser en particulier la signalisation ne rencontrent pas un déploiement massif. Comme cela a été précisé dans l'introduction, les limitations sont nombreuses : difficulté d'utiliser les certificats, problème d'interopérabilité, solution liée à un choix de protocole dans les couches supports (niveau transport ou routage), difficulté d'échanger des clés symétriques, absence de politique de sécurité universelle. Ce constat nous a conduit à définir le cahier des charges suivant pour permettre à une solution de sécurité pour la signalisation SIP d'émerger rapidement :

- Ne pas modifier le protocole SIP ;
- Ne pas ajouter de nouveaux champs dans les messages SIP ;
- Renforcer les mécanismes classiques présents.

Avec de telles contraintes, les évolutions restent possibles. Les en-têtes SIP comportent des valeurs opaques qui sont principalement générées de manière aléatoire pour le suivi des appels. Il a donc été envisagé d'utiliser ces champs en leur donnant une sémantique et par là même une propriété de sécurité.
Notre analyse a permis d'identifier un mécanisme de sécurité dans les messages SIP utilisant des valeurs opaques et pouvant bénéficier d'un renforcement : l'authentification HTTP Digest [5]. Ce mécanisme permet à un client SIP de s'authentifier auprès d'un serveur SIP mais pas l'inverse de manière native dans SIP pour des raisons de compatibilité avec la version antérieure à HTTP Digest. Ce constat peut permettre à un attaquant d'usurper l'identité d'un serveur légitime. L'attaque permet de réaliser un déni de service, la redirection des appels vers un numéro surtaxé ou encore la connaissance des destinataires de tous les appels. Notre solution vise donc à diminuer cette menace.

3. Authentification HTTP Digest SIP renforcée

3.1. Solution

L'authentification HTTP Digest [5] est un mécanisme basé sur un challenge/réponse. Il permet au client SIP de s'enregistrer auprès d'un fournisseur de service de téléphonie et d'avoir accès aux différentes ressources : l'authentification est par ailleurs demandée pour chaque requête SIP. Le serveur envoie un challenge au client, ce dernier répond par une valeur dérivée de ce challenge et d'un secret qu'il partage avec le serveur et généralement fourni avec le login par l'opérateur. Le serveur s'assure alors que le client possède effectivement le secret en calculant à son tour la réponse et en vérifiant la cohérence des deux.

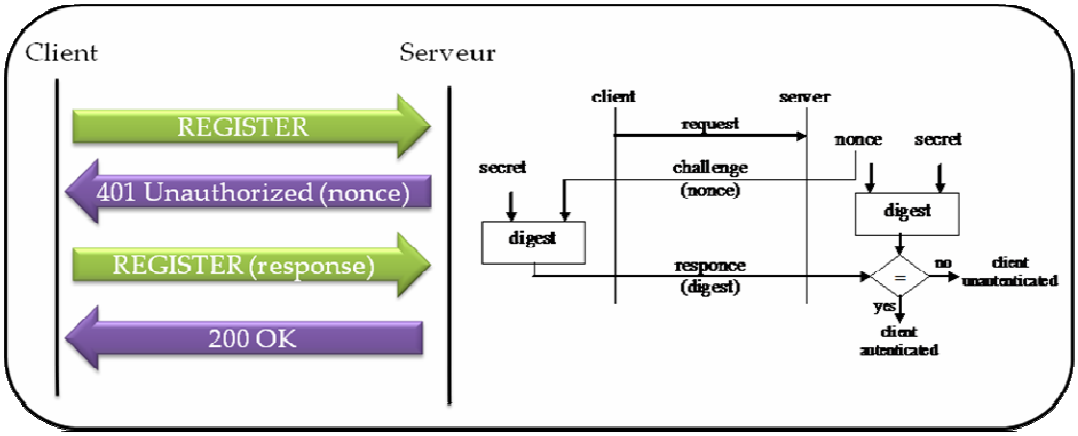


FIG. 3 - Echanges de messages SIP pour l'enregistrement d'un client avec le principe de l'authentification.

Les échanges de messages pour un enregistrement et le principe de l'authentification sont illustrés dans la figure 3. Le premier message informe le serveur du souhait du client de s'enregistrer par l'envoi d'une requête REGISTER. La réponse « 100 Trying » est un message d'attente, et la réponse « 401 Unauthorized » permet au serveur d'envoyer son challenge sous la forme d'un « nonce » inclus dans le message SIP. Le client calcule la réponse « response » avec le secret pré-partagé qui renvoie dans une nouvelle requête REGISTER. Si la valeur « response » est conforme à l'attente du serveur, ce dernier envoie donc une réponse « 200 Ok ». Le client est enregistré, il peut donc téléphoner mais il n'a aucune certitude que ce soit le serveur légitime avec lequel il dialogue.

Indépendamment d'une possible usurpation d'identité du serveur, l'authentification HTTP Digest peut présenter des faiblesses de sécurité. La principale faille réside dans la protection du mot de passe. Dans la mesure où le mode de calcul de la réponse est connu, un mot de passe peut faire l'objet d'une attaque force brute. Il convient donc d'avoir un secret le plus long possible.

L'authentification est incluse dans la syntaxe des messages SIP. Au message d'enregistrement REGISTER du client, le serveur répond par le message suivant en insérant le champ « WWW-Authenticate » qui contient le « nonce » :

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 137.194.192.237:5060; received=137.194.192.237
From: <sip:ahmed@enst.fr>
To: <sip:ahmed@enst.fr>;tag=as7b4af592
Call-ID: D8A5240D579C4D6E8CE1@enst.fr
CSeq: 7168 REGISTER
User-Agent: Asterisk PBX
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER
Max-Forwards: 70
Contact: <sip:ahmed@137.194.192.228>
WWW-Authenticate: Digest realm="asterisk", nonce="64d45b88"
Content-Length: 0
```

Champ précisant le type de message SIP

Champ précisant l'usager SIP concerné

Champ contenant les éléments générés par le serveur pour l'authentification de l'usager SIP

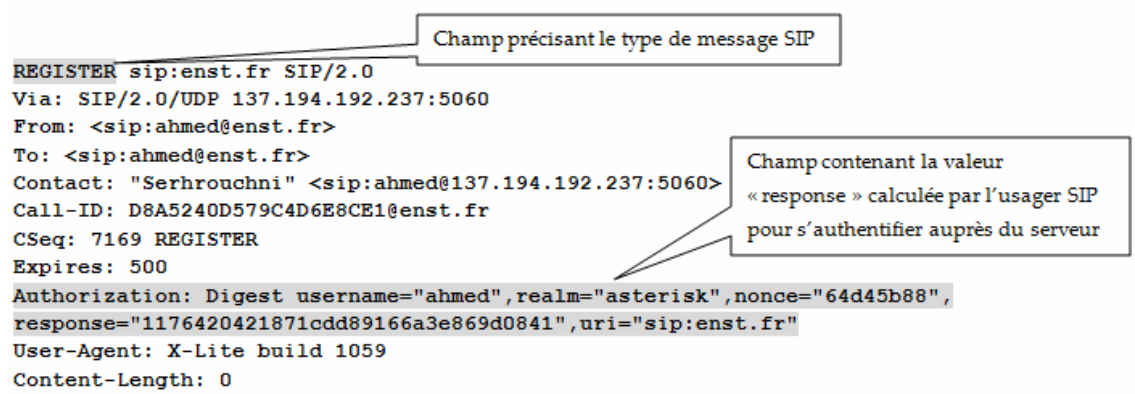
Le client reçoit donc un challenge dans le message dit "401 Unauthorized » sous la forme du « nonce ». Il forge alors la réponse en appliquant la formule suivante :

- *response=H(H(username | realm | password) | nonce | H(METHOD | Request-URI)) ;*
- *H est par défaut la fonction de hachage MD5 [11] comme spécifié dans [3].*

L'application à notre exemple donne :

- *response=H(H(ahmed | asterisk | password) | 64d45b88 | H(REGISTER | sip:enst.fr)).*

Le client renvoie un REGISTER avec un champ « Authorization » et la valeur « response » :



Notre cahier des charges peut donc s'appliquer au « nonce ». Ce dernier est généré de manière aléatoire et opaque selon les spécifications de [5]. Notre proposition pour l'élaboration du « nonce » est la suivante :

{nonce =H(H(username | realm | password) | callid-value)}

La première partie du calcul reprend une partie de la sémantique de « response ». Par ailleurs le « realm » doit devenir une valeur aléatoire pour éviter les problèmes de rejeu, ce qui est possible compte tenu de son rôle et de ces spécifications : en effet si le nonce dépend uniquement du call-id, un pirate pourrait intercepter cette valeur et la rejouer, connaissant ainsi déjà la valeur « response ». Par défaut, la fonction de hachage est MD5 comme pour une authentification HTTP Digest SIP.

L'authentification du serveur par le client est obtenue par la vérification du nonce de la même manière que le serveur vérifie la valeur « response ». La figure 4 résume la proposition. Cette approche est valable pour les autres requêtes SIP, une authentification étant demandée pour chaque demande de ressources. Appliqué à l'exemple ci-dessous, le nonce calculé est :

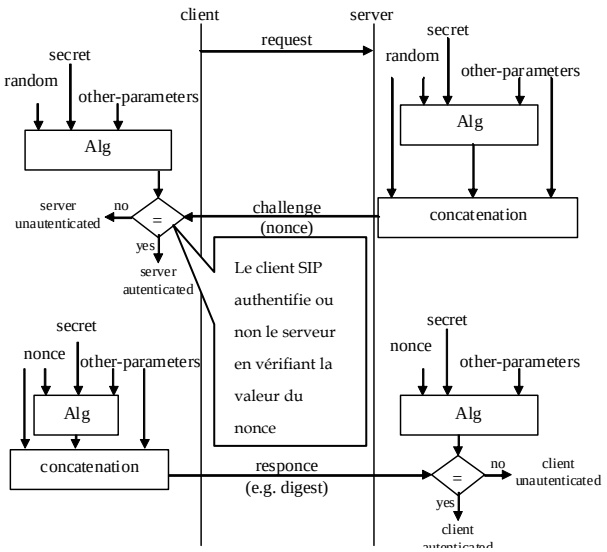


FIG. 4 - Principe de notre proposition.

- *nonce=H(H(ahmed | asterisk | password) | D8A5240D579C4D6E8CE1@enst.fr) ;*
- *nonce= fa4112d21f9ba263c6fd197e6850f6c4.*

Pour faciliter le déploiement, l'authentification du serveur doit être envisagée comme un service. La réussite ou l'échec de la procédure doit être remonté au client sous forme d'une information. Ce choix permet de rendre complètement compatible les équipements avec ou sans la solution de renforcement de l'authentification HTTP Digest.

3.2. Validation formelle avec l'outil AVISPA

Avant de l'implémenter, la solution a été validée formellement par l'outil de sécurité AVISPA [1] (Automated Validation of Internet Security Protocols and Applications). Ce programme est maintenant couramment utilisé pour valider les nouveaux protocoles. AVISPA utilise un langage High Level Protocol Specification Language (HLPSL) pour décrire les protocoles. Les protocoles ainsi spécifiés (cf. annexe 1) sont ensuite injectés dans des modules d'analyse pour vérifier leurs propriétés.

Notre analyse a donc commencé par formaliser en HLPSL une authentification HTTP Digest SIP standard et une authentification HTTP Digest SIP renforcée selon nos spécifications (les spécifications HLPSL de notre solution sont données annexe I). Un contrôle avec le module « On-the-Fly Model-Checker » a confirmé notre analyse sur l'apport de notre solution en la qualifiant de SAFE ; l'authentification du serveur permettant la disparition de l'attaque de « l'homme au milieu » trouvé dans le cas de l'authentification HTTP Digest (cf. tableau 1). L'analyse du modèle ne prend pas en compte les éventuelles faiblesses des algorithmes cryptographiques.

Résultat avec une authentification HTTP Diget	Résultat avec notre proposition
SUMMARY UNSAFE DETAILS ATTACK_FOUND PROTOCOL HTTP DIGEST ... OFMC ... ATTACK TRACE i ->(ss,3): sipregister.uac (ss,3) -> i: sip401.Nonce(1) i ->(uac,3): start (uac,3) -> i: sipregister.uac i->(uac,3):sip401.Nonce(1)(uac,3)>i:sipregister.uac.Nonce(1).h(Nonce(1).h(uac.pwd)) i->(ss,3): sipregister.uac.Nonce(1).h(Nonce(1).h(uac.pwd))(ss,3) -> i: sip200	SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL SIP MUTUAL AUTHENTICATION GOAL as_specified BACKEND OFMC COMMENTS STATISTICS ...

Evaluation globale du protocole par AVISPA

Evaluation globale du protocole par AVISPA

Vulnérabilité trouvée par AVISPA

TAB. 1 - Comparaison entre l'authentification HTTP Digest et notre proposition avec l'outil AVISPA.

La validation avec AVISPA confirme l'intérêt de donner une sémantique au « nonce » pour renforcer l'authentification HTTP Digest SIP.

3.3. Implémentation sur une plate forme logicielle utilisant ASTERISK

Une implémentation sur une plate-forme logicielle a été réalisée pour tester notre solution (cf. figure 5). Elle était constituée d'un IPBX Asterisk [2] qui est un logiciel de traitement d'appels Open source supportant le protocole SIP et du softphone SIP Twinkle [20]. Des tests ont été réalisés en alternant l'emploi des logiciels originaux et ceux modifiés pour intégrer notre solution. Le tableau 2 reprend l'ensemble des situations validées.

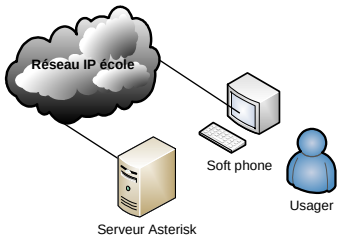


FIG. 5 - Plate-forme de validation.

Les tests montrent que l'intégration du nonce avec sémantique n'impacte pas le fonctionnement des équipements modifiés. Cet ajout est tout à fait transparent pour les équipements qui ne supportent pas le nonce calculé. Si Asterisk n'est pas modifié avec notre solution, l'usager constate l'absence de l'information « Serveur authentifié » : la décision d'utiliser cette IPBX pour traiter ses appels lui revient. Si c'est le client qui ne supporte pas notre nonce, il calcule la valeur « response » pour s'authentifier sans modifier son fonctionnement.

Client Twinkle normal	Client Twinkle modifié	IPBX Asterisk normal	IPBX Asterisk modifié	Comportements des équipements dans les différentes configurations possibles
✓		✓		Situation normale. Le client s'authentifie auprès du serveur.
✓			✓	L'IPBX génère un nonce avec sémantique mais le client ne l'exploite pas. Il s'authentifie néanmoins auprès du serveur en fournissant un « response » calculé à partir du nonce.
	✓	✓		L'IPBX génère un nonce de manière aléatoire. Le client constate que le nonce n'a pas de sémantique. Il est informé de cette situation par un message « server unauthenticated ». Le client fait alors le choix d'utiliser ou non l'IPBX pour téléphoner.
	✓		✓	L'IPBX s'authentifie auprès du client en générant un nonce avec sémantique. Le client est informé par le message « server authenticated ». Ensuite il s'authentifie auprès du serveur en fournissant un « response » calculé à partir du nonce.
✓ : équipements utilisés.				

TAB. 2 – Résultats des tests sur la plate-forme.

4. Conclusion

La sécurité des systèmes d'informations devient chaque jour un peu plus prégnante, alors même que ces systèmes deviennent plus complexes et plus vulnérables aux menaces. Tous les efforts doivent être entrepris pour renforcer la téléphonie sur IP. La confiance des usagers et des opérateurs est à ce prix.
L'hétérogénéité des implémentations et des architectures réseaux ne permet pas d'avoir une solution standard. Le renforcement de la sécurité au travers d'un service, tout en préservant l'interopérabilité, peut être considéré comme une valeur ajoutée au standard. Nos futurs travaux s'attacheront à intégrer d'autres mécanismes de sécurité basés sur les valeurs opaques de SIP comme l'utilisation des mots de passe à usage unique.

Bibliographie

1. A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. Hankes Drielsma, P.C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò and L. Vigneron. The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications". *Computer Aided Verification Book*, Spinger, volume 3576/2005: 281-285, juillet 2005.
2. Projet Asterisk : http://www.asterisk.org.
3. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler. SIP : Session Initiation Protocol. *RFC 3261*, juin 2002.
4. T. Dierks et C. Allen. The TLS Protocol. *RFC 2246*, janvier 1999.
5. J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen and L. Stewart, HTTP Authentication: Basic and Digest Access Authentication. RFC 2617, June 1999.
6. Systèmes de communication multimédia en mode paquet. *Recommandation UIT-T H.323*, juin 2006.
7. S. Kent, and R. Atkinson. Security Architecture for the Internet Protocol. *RFC 2401*, novembre 1998.
8. J. Postel. User Datagram Protocol. *RFC 768*, août 1980.
9. J. Postel. Transmission Control Protocol STD 7. *RFC 793*, septembre 1981.

10. B. Ramsdell. Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification. *RFC 3851*, juillet 2004.
11. R. Rivest. The MD5 Message-Digest Algorithm. *RFC 1321*, avril 1992.
12. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach et T. Berners-Lee. Hypertext Transfer Protocol -- HTTP/1.1. *RFC 2616*, juin 1999.
13. P. Gupta, V. Shmatikov. Security Analysis of Voice-over-IP Protocols. *Computer Security Foundations Symposium*, 2007, CSF '07, 20th IEEE, Venice, p 49-63, juillet 2007.
14. El Sawda and P. Urien. SIP Security Attacks and Solutions : A state-of-the-art review. *Information and Communication Technologies 2006*, ICTTA'06, Damascus, vol.2 : 3187–3191, avril 2006.
15. H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. *RFC 3550*, juillet 2003.
16. F. Chabaud. Recherche et développement en sécurité des systèmes d'information : orientations et enjeux. *Actes de la conférence SSTIC*, Rennes, juin 2008.
17. F. Yergeau. UTF-8, a transformation format of ISO 10646. *RFC 2279*, janvier 1998.
18. M. Zandi, M.V. Martin and P.C.K. Hung. Overview of security issues of VoIP. *Proceeding of IASTED European Conference Internet and Multimedia Systems and Applications*, Chamonix, mars 2007.
19. Logiciel et services Skype : http://www.skype.com.
20. Softphone Twinkle: http://www.xs4all.nl/~mfnboer/twinkle/index.html.

Annexe 1 : Spécification en HLPSL de l'authentification HTTP Digest SIP renforcée

```
role sip_server(SS,UAC : agent, PWD : text, H :
hash_func, SND, RCV : channel(dy))
played_by SS def=
local State : nat,
Callid, Realm : text
init State := 1 transition
0. State = 1
/\ RCV(sipregister.UAC'.Callid') =|> State' := 2
/\ Realm' := new()
/\
SND(sip401.UAC.Callid.Realm'.H(H(UAC.Realm'.PW
D).Callid))
/\ wit-
ness(SS,UAC,yy,H(H(UAC.Realm'.PWD).Callid))
1. State = 2
/\RCV(sipregister.UAC.Callid.H(H(H(UAC.Realm.P
WD).Callid').PWD.UAC)) =|> State' := 3
/\ SND(sip200)
/\request(SS,UAC,y,H(H(H(UAC.Realm.PWD).Callid
).PWD.UAC))
end role
role user_agent_client(UAC,SS : agent, PWD : text, H :
hash_func, SND, RCV :channel(dy))
played_by UAC def=
local State : nat,
Callid, Realm : text
init State := 1
transition
2. State = 1
/\ RCV(start) =|> State':=2
/\ Callid' := new()
/\ SND(sipregister.UAC.Callid')
3. State = 2
/\RCV(sip401.UAC.Callid.Realm'.H(H(UAC.Realm'.P
WD).Callid)) =|> State':= 3
/\SND(sipregister.UAC.Callid.H(H(H(UAC.Realm.P
WD).Callid).PWD.UAC))
/\witness(UAC,SS,y,H(H(H(UAC.Realm.PWD).Callid
).PWD.UAC))
/\request(UAC,SS,yy,H(H(UAC.Realm.PWD).Callid))
4. State = 3
/\ RCV(sip200) =|> State':= 4
end role
role session(UAC,SS:agent, H:hash_func, PWD:text)
def= local SND, RCV : channel(dy)
composition
sip_server(SS,UAC,PWD,H,SND,RCV)
/\user_agent_client(UAC,SS,PWD,H,SND,RCV)
end role
role environment() def=
const uac, ss : agent,
h : hash_func,
y,yy,sip401, sip200, sipregister : protocol_id,
pwd : text
intruder_knowledge=
{uac,ss,sipregister,sip401,sip200,h,i}
composition session(uac,ss,h,pwd)
end role
goal
authentication_on y
authentication_on yy
end goal
environment()
environment()
```