Livre Blanc du protocole SILC

Pekka Riikonen - Projet SILC

Version 1.2 / 22 Octobre 2003
Traduction : Version 1.0 / Juillet 2010

I. Introduction

Les protocoles de chat sur Internet sont très populaires. Ils le sont depuis l'apparition des tous premiers protocoles de chat sur Internet. L'IRC (Internet Relay Chat - Chat relayé par Internet) fut l'un des premiers protocoles de chat, et gagna rapidement le statut de chat le plus populaire sur le net. Aujourd'hui, IRC est en concurrence avec de nombreux autres protocoles de Messagerie Instantanée (Instant Messaging - IM), comme ICQ. Cependant, ils possèdent tous un point commun : aucun n'est sécurisé !

De nos jours, la sécurité est une fonctionnalité importante des applications et protocoles dans un environnement réseau. Les plus vieux protocoles de chat, n'ont cependant pas réussi à remplir les besoins croissants en sécurité sur Internet. Il n'est plus suffisant de fournir des services tels que les services de chat. Ils doivent aussi être sécurisés.
Le protocole de conversation sécurisée en directe par internet (SILC) est un protocole de nouvelle génération qui fournit des services complets de conversation, de la même manière que les autres protocoles actuels de chat. De plus, il est sécurisé par le cryptage et l'authentification des messages sur le réseau. Le premier objectif du protocole SILC a été la sécurité, et il a été développé dans le besoin actuel de sécurité. Tous les paquets et messages circulant sur le réseau SILC sont toujours cryptés et authentifiés. La topologie du réseau est différente des autres réseaux comme IRC par exemple. La topologie du réseau SILC est plus puissante et évolutive
que celle du réseau IRC. Le but premier du protocole SILC est de founir des services de conversation sécurisés.
Le protocole SILC a été développé en tant que projet Open Source. les spécifications du protocole sont librement disponibles et ont été envoyées à l'IETF. le protocole est actuellement en cours de stabilisation et a atteint sa version 1.2.

II. À propos de ce libre blanc

Le but de ce livre blanc est de donner une courte introduction mais assez détaillée du protocole SILC. Ce document décrit l'objectif du protocole et son fonctionnement pratique. Il est accessible à tous les publics. Ce papier devrait être facile à comprendre pour des gens sans connaissances techniques, mais assez détaillé pour une personne maîtrisant le sujet. Lisez la section "Termes et abréviations" pour les termes techniques utilisés dans ce document.

III. Le protocole SILC

Discussion en directe sécurisée sur internet (Secure Internet Live Conferencing - SILC), ou SILC en abrégé, est un protocole moderne de discussion qui fournit un grand nombre de fonctionnalités pour la discussion, et ce avec une sécurité élevée. Un des principaux principes de conception du protocole a été la sécurité. Beaucoup des fonctionnalités de SILC sont trouvables dans les protocoles de chat traditionnels comme IRC ou dans les protocoles du type messagerie instantanée (Instant Message -IM).
SILC combine aussi les fonctionnalités de ces deux types de protocoles de chat, et peut être

implémenté de manière à posséder les fonctions de l'un ou de l'autre? En fait, SILC supprime le besoin de faire la distinction entre ces deux types de protocoles. Certaines des fonctionnalités les plus avancées, et celles liées à la sécurité sont nouvelles à tous les protocoles de discussion. SILC supporte aussi les messages multimédia et peut donc aussi être implémenté en tant que système de discussion audio et vidéo. Le protocole est compact et robuste et s'adapte bien aux environnements mobiles où la faible bande passante établit des exigences particulières pour les protocoles. Toutes les tailles de paquet dans SILC peuvent encore être réduites par compression.

Les paquets et messages sur le réseau SILC son toujours cryptés et authentifiés. Il est impossible d'envoyer des messages non cryptés dans SILC. Cela donne l'assurance que l'utilisateur final ne peut pas envoyer accidentellement un message non crypté en pensant qu'il l'est. C'est l'un des problèmes de la plupart des protocoles de chat fournissant des plug-in de cryptage. Ils ne sont pas sécurisés par défaut, mais ils essayent de fournir de la sécurité en appliquant des protocoles de sécurité externes tels que PGP ou SSL sur un protocole pas sécurisé. Dans ces cas, la sécurité est habituellement atteinte en cryptant les données puisque le gestionnaire de clés, l'authentification du message et les autres problèmes de sécurité sont mis de côté, laissant l'implémentation vulnérable à divers problèmes de sécurité. L'autre problème est que les protocoles externes tendent à laisser le réseau seulement sécurisée de manière partielle; d'habitude, seuls deux points d'un réseau sont sécurisés, comme dans SSL, par exemple. Bien que SSL fournisse une sécurité certaine, ce n'est pas assez pour fournir de la sécurité à un réseau de chat dans son ensemble.

SILC reste sécurisé dans un environnement de méfiance mutuelle entre deux entités du réseau. Il est possible de crypter des messages d'un bout à l'autre, pour que seul l'expéditeur et le destinataire soient capables de crypter et décrypter les messages. Il est tout-à-fait possible d'envoyer des messages à un groupe d'utilisateurs, de manière à ce que seul le groupe concerné soit capable de crypter et décrypter les messages. Souvent, le protocole utilise des clés qui sont générées par les serveurs, ***afin que si le mode d'échange des clés tombe le réseau reste crypté***. Cependant, il est toujours possible d'échanger et d'utiliser des clés générées localement pour sécuriser les messages, et ainsi le serveur ne connait pas les clés.

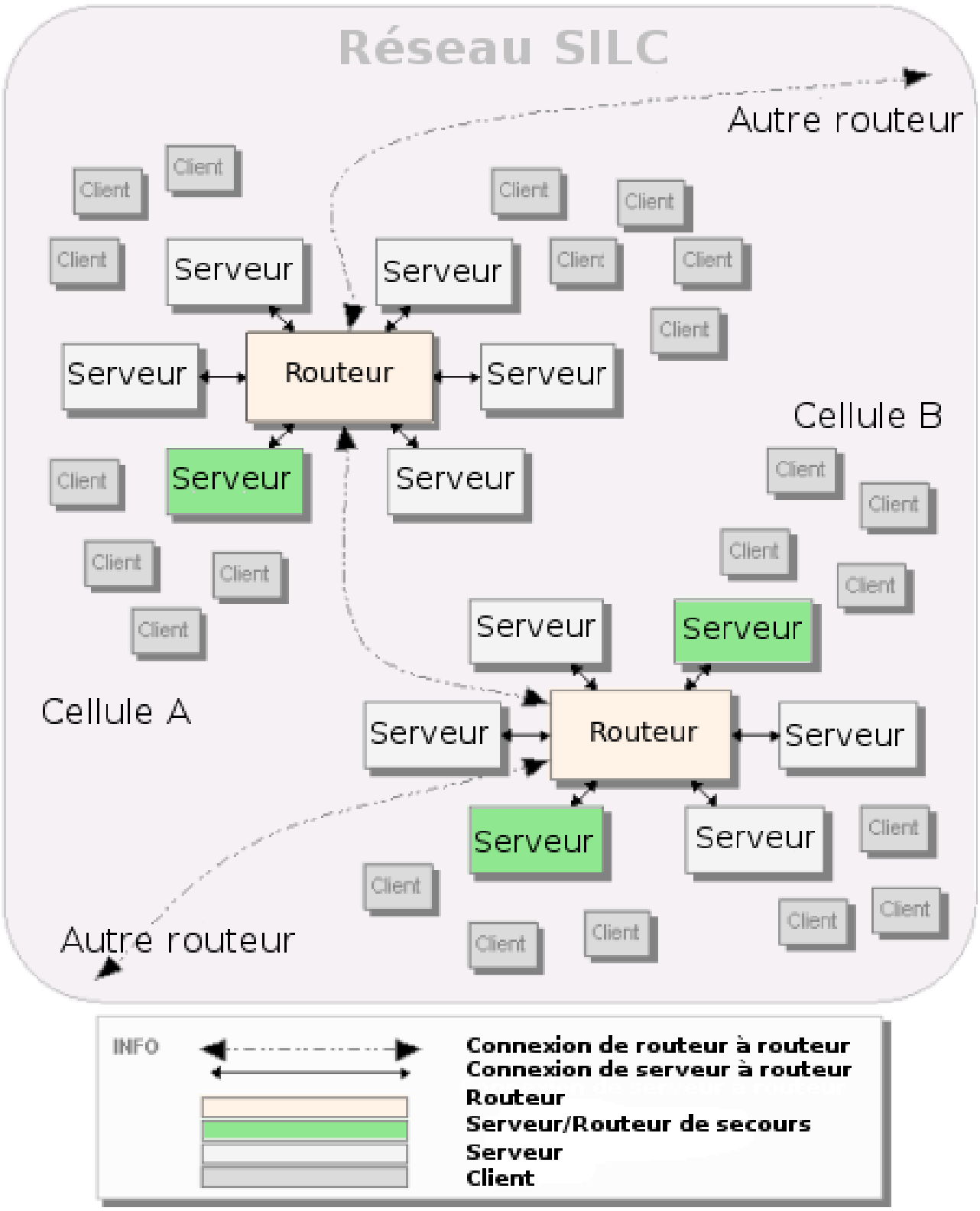
Comme la plupart des protocoles de chat actuels, SILC permet le transfert de fichiers. Il est possible de transférer des fichiers de manière sécurisée entre des utilisateurs du réseau SILC. ***Le flux de transfert de fichier est toujours envoyé vers l'extérieur du réseau de pair à pair***. Avant que le transfert de fichier commence, un protocole d'échange de clé est exécuté pour échanger la clé de session pour le transfert de fichier.

Le protocole SILC supporte aussi le détachement, une nouvelle idée qui permet de se détacher d'un serveur sans vraiment quitter le réseau. Il est alors possible de reprendre la connexion sur un des serveurs du réseau, comme si vous n'étiez jamais parti. Le protocole SILC rend en outre possible la distribution et l'échange de clés publiques et des certificats par le réseau SILC. On peut aussi recueillir des informations détaillées à propos d'autres utilisateurs du réseau SILC. On peut par exemple prendre la carte de visite d'un utilisateur, sa photo, ses certificats, etc. Le protocole SILC supporte le transfert sécurisé de fichier afin de permettre l'échange de fichier sécurisé entre les utilisateurs.

Le protocole SILC supporte aussi les services fournis par des extensions du protocole d'origine. Elles peuvent être utilisées pour augmenter les fonctionnalités du protocole ou pour ajouter en ajouter de nouvelles sans rompre la compatibilité avec les versions antérieures. Les services peuvent être échangés en ligne et authentifiés avec des phrases de passe ou des signatures numériques.

La topologie du réseau est différente de celle des traditionnels protocoles de chat et de discussion. Le réseau SILC forme un réseau hybride maillé en anneau au niveau des routeurs, et un réseau en

étoile au niveau des serveurs. Ce type de topologie de réseau permet une meilleure extensibilité et une plus rapide délivrance des paquets que sur un réseau en arbre. Les serveurs de type routeur sont distincts des serveurs normaux car seuls les routeurs connaissent les informations générales du réseau et le gardent à jour, les serveurs normaux, eux, conservent seulement à jour les informations locales. Cela améliore notablement l'extensibilité du réseau. Le réseau supporte aussi les routeurs de secours qui peuvent être utilisés afin de lutter contre les coupures de connexion.



Le diagramme ci-dessus représente une portion du réseau SILC. Il montre deux cellules qui elles-mêmes possèdent des routeurs de secours et de nombreux serveurs et clients. Les clients peuvent se connecter aux serveurs, et aux routeurs s'ils le désirent. Les sections suivantes décriront plus en détail les entités composant le réseau SILC.

III.1. Les clients

Un client est un morceau de programme se connectant à un serveur SILC. Le programme est habituellement lancé par un utilisateur final, c'est-à-dire une personne réelle. Le but des clients est de fournir à l'utilisateur final une interface permettant l'usage des services SILC. En effet, Ils sont utilisés pour engager des discussions sur le réseau SILC, et ils peuvent servir à lancer diverses commandes SILC.

Chaque client se différencie des autres grâce à un ID client unique. Il ne peut pas y avoir simultanément plusieurs ID client identiques sur le réseau SILC . L'utilisateur final, ne se sert pas de ces ID. Les utilisateurs finaux choisissent la plupart du temps un pseudonyme qu'ils désirent utiliser, et ils se reconnaissent entre eux par le biais de ces pseudonymes. Il peut y avoir simultanément plusieurs pseudonymes identiques sur le réseau SILC. La taille maximum d'un pseudonyme est de 128 bytes.

La majeure partie des autres protocoles de chat se servent de pseudonymes uniques. Sur ce point, SILC est différent. Le but de cette fonctionnalité est d'empêcher une inutile guerre de pseudonymes comme il peut en exister sous IRC, puisque personne ne possède son propre pseudonyme; il pourra toujours y avoir quelqu'un d'autre avec le même pseudonyme. Cette fonctionnalité rend aussi les services d'enregistrement des pseudonymes obsolètes.

Quand un client se connecte à un serveur SILC, les protocoles d'échange de clé (SILC Key Exchange - SKE) et d'authentification de connexion sont lancés. La clé de session est produite par le SKE, et permet au client et au serveur de sécuriser leur communication. Par exemple, Toutes les commandes que le client envoie au serveur sont sécurisées avec cette clé. Elle expire régulièrement et le processus de génération de clé peut-être exécuté avec ou sans le PFS (Perfect Forward Secrecy). Le protocole d'authentification de la connexion permet l'authentification du client par le serveur. Le serveur peut tout aussi bien permettre au client de se connecter sans authentification, ou en demandant une phrase de passe ou encore une authentification basée sur une clé publique (ou un certificat).

III.2. Les serveurs

Les serveurs forment la base du réseau SILC puisqu'ils fournissent un point depuis lequel les clients peuvent se connecter. Il existe deux types de serveurs : les normaux et les routeurs. La section suivante décrit le fonctionnement d'un routeur.

Les serveurs se connectent aux routeurs. Les serveurs ne peuvent pas se connecter entre eux directement. Les messages qui sont pour l'extérieur du serveur local sont toujours envoyés aux routeurs qui les redirigent plus loin. Les client se connectent en général aux serveurs, cependant, les clients peuvent aussi directement se connecter aux routeurs. Le diagramme du réseau SILC ci-dessus illustre la manière dont les serveurs se connectent aux routeurs.

Les serveurs se distinguent entre eux sur le réseau par l'attribution d'un ID de serveur unique. Il ne peut y avoir simultanément deux IDs de serveurs identiques sur le réseau SILC. Les serveurs

suivent l'évolution des informations locales. Il connaît tous les clients locaux connectés ainsi que tous les canaux que ces clients ont joints. Cependant, il n'a pas connaissance des informations globales. En général, il ne suit pas l'actualité globale des clients, en revanche, il peut mettre en cache ces informations si cela a été demandé. La raison est que cela permet aux serveurs de ne pas chercher à mettre à jour les informations générales du réseau, et permet donc d'améliorer la rapidité du serveur (et par ce biais, la rapidité du réseau entier). Ils peuvent toujours avoir ces informations via le routeur.

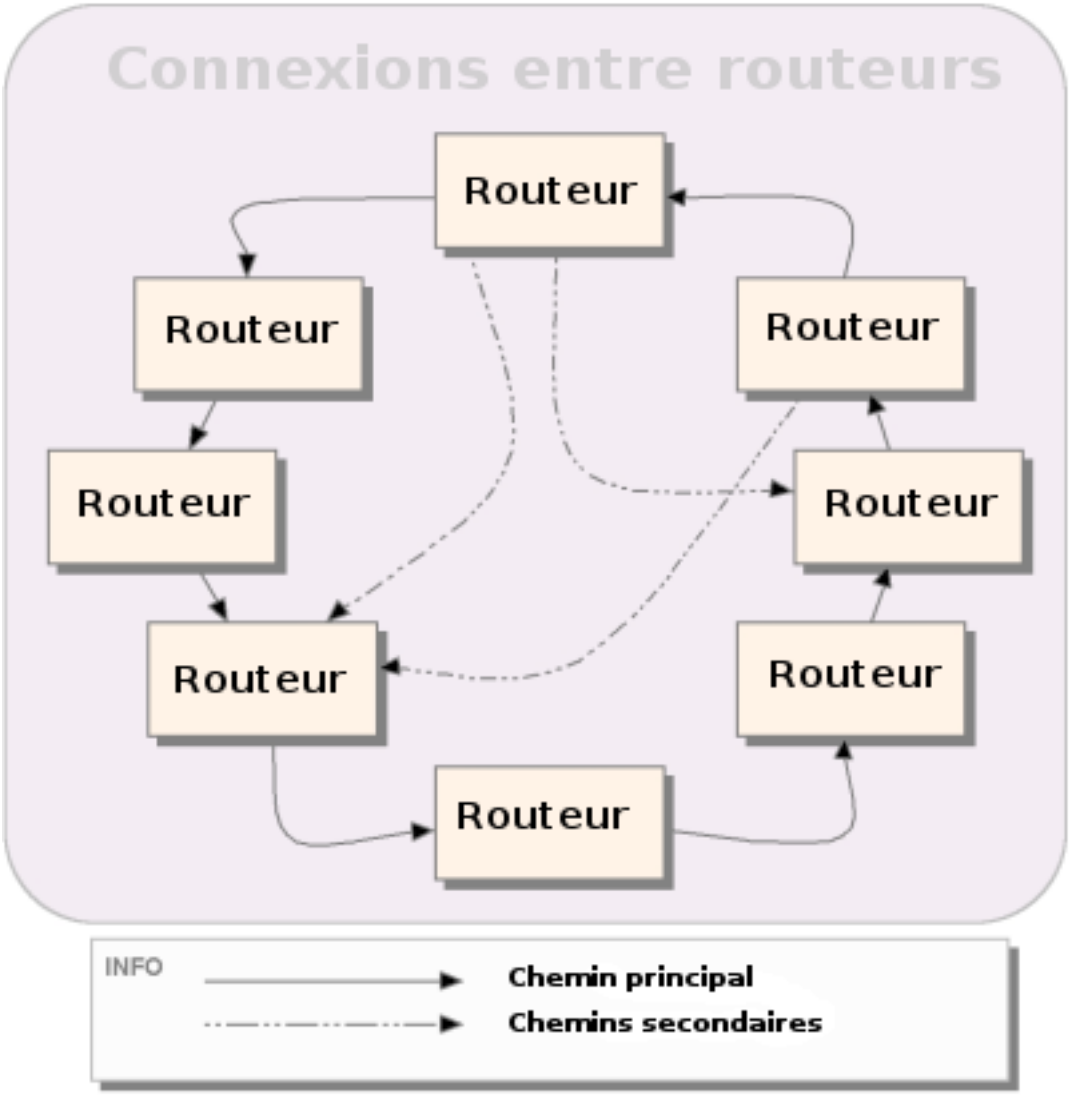
Quand un serveur se connecte à son routeur, le protocole d'échange de clés SILC (SILC Key Exchange - SKE) et le protocole d'authentification des connexions SILC sont lancés, de la même manière que lors de la connexion d'un client à un serveur. Le SKE renvoie la clé de session qui sert à sécuriser la communication entre le serveur et le routeur. Le protocole d'authentification de la connexion permet au serveur de s'authentifier auprès du routeur. L'authentification est toujours basée sur une phrase de passe ou une clé publique (ou un certificat).

III.3. Les routeurs

Les routeurs sont des serveurs qui s'occupent du routage des messages sur le réseau. Ils se comportent aussi comme des serveurs normaux et acceptent les connexions de clients. Chaque routeur du réseau est appelé une cellule. Celle-ci peut contenir un seul routeur actif, mais plusieurs serveurs et clients. La cellule peut néanmoins posséder des routeurs de secours qui puissent prendre le relais du routeur principal si celui-ci ne répond plus. Le passage au routeur de secours doit être transparent, et seules les connexions locales sont perdues. Les autres connexions dans la cellule sont intactes, et les clients et serveurs subiront juste quelques lags dans la connexion au réseau durant le passage sur le routeur de secours.

Les serveurs normaux connaissent seulement les informations locales. Les routeurs, quand à eux, connaissent les informations locales et globales. Ils considèrent comme local ce qui est à l'intérieur de la cellule, et global ce qui y est extérieur. Ils connaissent tous les client connectés, tous les canaux créés, et tous les routeurs et serveurs du réseau. Le serveur peut récupérer les informations globales si nécessaire. Quand un client envoie une commande WHOIS par exemple, le serveur peut récupérer l'information du routeur. Si le routeur ne connait pas tous les détails demandés par la commande WHOIS, il peut récupérer ces informations d'un routeur ou d'un serveur les connaissant. Il peut ensuite mettre en cache ces informations.

Le premier objectif du routeur est d'acheminer les messages aux serveurs et clients locaux, et aussi ceux destinés à l'extérieur de la cellule par le chemin principal ou par le chemin secondaire s'il est plus rapide. Les routeurs du réseau forment un anneau. Chaque routeur possède un chemin principal vers un autre routeur du réseau. Finalement, l'anneau est fermé par le dernier routeur qui utilise le premier comme chemin principal.



Le diagramme ci-dessus illustre la façon dont les routeurs forment un anneau dans le réseau. Un routeur peut avoir plusieurs chemins secondaires qu'il peut utiliser pour acheminer les paquets.

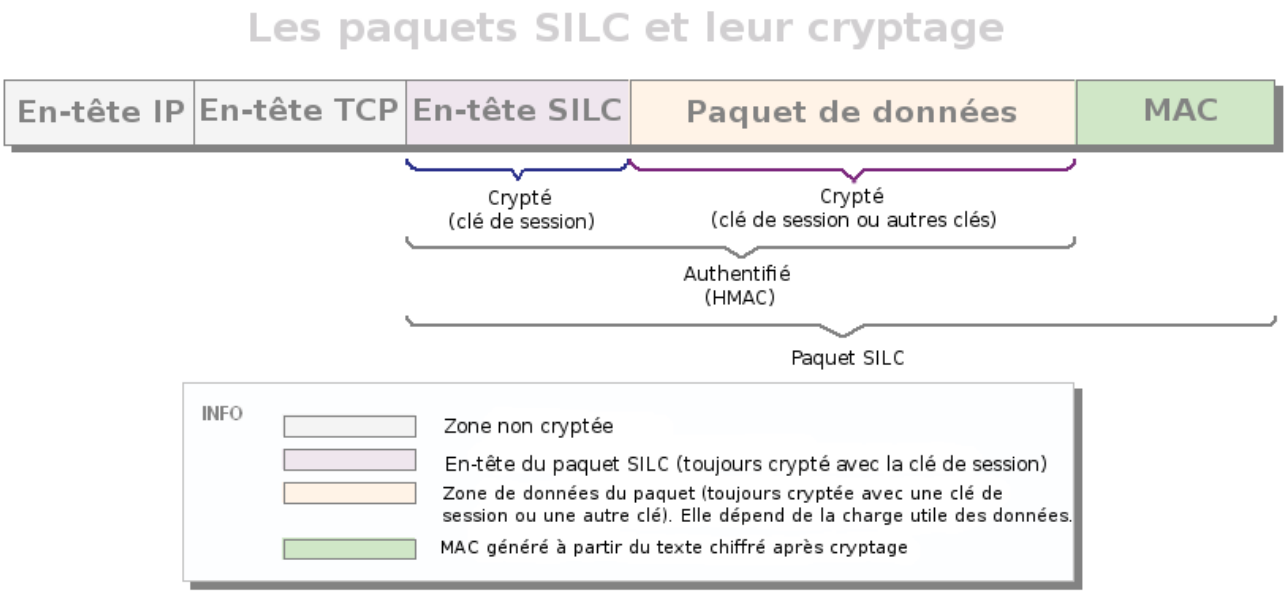
Quand des routeurs se connectent à leur routeur principal, les protocoles de SKE et d'authentification de connexion SILC sont lancés, tout se passe comme si un serveur se connectait à un routeur. La clé de session est utilisée pour sécurisée la communication entre les routeurs. Tous les chemins secondaires ont leur propres clés de session.

IV. Le protocole de paquets SILC

La base du protocole SILC repose sur les paquets SILC qui sont, sans aucun doute, la partie la plus importante du protocole. Le protocole de paquets SILC est un protocole sécurisé de paquet binaire. Ce dernier fournit des paquets binaires sécurisés et assurent que leur contenu soit authentifié et sécurisé.

Les paquets sont utilisés en permanence par le protocole SILC, que ce soit pour envoyer des

messages de canal, des messages privés, des commandes ou d'autres informations. Tous les paquets circulant sur le réseau SILC sont cryptés et leur intégrité est assurée par le calcul de Codes d'Authentification de Message (Message Authentication Codes - MAC). Le protocole définit plusieurs types de paquets et leur charge utile. Chaque type de paquet possède sa propre charge utile qui définit le contenu du paquet. Les données réelles du paquet correspondent donc aux charges utiles définies dans le protocole.



Comme le diagramme ci-dessus l'illustre, le paquet SILC est constitué de l'en-tête de paquet SILC, d'une zone de données qui inclut la charge utile du paquet et d'une zone MAC qui assure l'intégrité du paquet. La totalité du paquet SILC est crypté, et ce en permanence, à l'exception de la zone MAC qui n'est jamais cryptée. Le processus de cryptage et la clé utilisée dépendent de la charge utile du paquet. Certaines charges utiles sont cryptées avec la clé de session, et des fois avec d'autres clés comme les clés de message de canal. L'en-tête du paquet SILC est toujours crypté avec la clé de session. Le MAC est calculé après le cryptage de l'en-tête du paquet SILC et de la zone de données. On appelle ce procédé l'ordre cryptage-puis-MAC (Encrypt-Then-MAC).

V. Le protocole d'échange de clé SILC

Le protocole d'échange de clé SILC (SILC Key Exchange Protocol - SKE) est utilisé pour échanger des données secrètes entre des entités connectées. Le résultat de ce protocole est une clé utilisée afin de sécuriser les canaux de communications. Il est lancé lors de la connexion d'un client à un serveur par exemple, ou d'un serveur à un routeur ou encore d'un client à un autre, si ces deux derniers désiraient créer une clé secrète. L'objectif de ce protocole SKE c'est de créer des clés de session utilisée dans les sessions SILC en cours. Le SKE est basé sur l'algorithme d'échange de clé de Diffie-hellman, et est immunisé face aux attaques du type de l'homme-du-milieu (man-in-the-middle) par l'emploi de signatures numériques.

Il s'agit du premier protocole lancé lors d'une création de connexion avec un serveur SILC, par exemple. Tous les autres protocoles se lancent après celui-là. Ainsi, tous les autres protocoles sont sécurisés puisque le SKE a créé la clé de session qui sert à sécuriser tous les paquets ultérieurs. La clé de session crée par le SKE n'est valide que pendant un certain laps de temps (une heure

habituellement) ou, au maximum, jusqu'à la fin de la session. Le processus de recréation de clé peut être lancé avec ou sans le Perfect Forward Secrecy (PFS).

Les fonctionnalités de sécurité qui sont utilisés dans la session SILC sont aussi négociées pendant le SKE. Le protocole possède un initiateur et un destinataire. L'initiateur est celui qui lance la négociation du SKE et le destinataire est celui qui répond à cette négociation du SKE. Quand le protocole est lancé, il envoie à l'initiateur une liste des fonctionnalités de sécurité qu'il supporte. Le destinataire choisit alors celles qu'il supporte et envoie sa réponse à l'initiateur. Les fonctionnalités de sécurité comprennent les chiffrements, les fonctions de hachage, les algorithmes à clé publique, les fonctions HMAC, etc. Le destinataire peut toujours choisir celles qu'il supporte.

Après la sélection des fonctionnalité de sécurité, le protocole exécute l'algorithme d'échange de clé de Diffie-Hellman. Au même moment, l'initiateur et le destinataire s'envoient mutuellement leur clé publique ou leur certificat. Ils calculent une signature que l'autre partie vérifiera. Le protocole est exécuté par défaut dans un mode appelé authentification mutuelle, où les deux parties calculent indépendamment une signature que l'autre vérifie. De cette manière les deux parties prouvent leur possession de la clé privée liée à la clé publique qu'ils ont fournis au protocole. En cas d'échec au cours d'une des étapes du protocole, la connexion est immédiatement arrêtée.

La clé publique ou le certificat reçu pendant le protocole de SKE doit être vérifié. S'il ne l'était pas, il serait possible d'exécuter une attaque man-in-the-middle sur le protocole de SKE. En cas d'utilisation de certificats, ces derniers doivent être vérifiés par une autorité de certification (Certification Authority - CA). La vérification d'une clé publique requiert la confirmation de l'empreinte de la clé publique par téléphone ou e-mail, ou bien le serveur peut, par exemple, publier l'empreinte (et la clée publique) sur quelques sites web. Dans la pratique, les systèmes acceptant la clé publique sans vérification sont souvent voulus. Dans beaucoup de protocoles de sécurité, tel que SSH2, la clé publique est acceptée sans vérification à la première connexion. Celle-ci est alors mise en cache sur le disque dur local. Lors de la prochaine connexion au serveur, la clé publique présente sur le disque local est comparée à celle envoyée par le serveur. Dans la réalité cela ne pose pas de problème la plupart du temps. Néanmoins, si un client (ou un serveur) ne peut pas lui faire confiance, il doit trouver une autre façon de vérifier la clé publique ou le certificat reçu.

VI. Protocole d'authentification de connexion SILC

Le but du protocole d'authentification de connexion SILC est d'authentifier la partie se connectant au serveur ou au routeur. Il est lancé lorsque des clients se connectent au serveur, par exemple. Il est aussi lancé lors de la connexion d'un serveur à un routeur. Son autre but est de fournir des informations au serveur sur le type de connexion. Le type de connexion définit s'il s'agit d'un client, d'un serveur, ou d'un routeur. Si c'est un client alors le serveur créera un nouvel ID client pour celui-ci. Dans le cas d'un serveur alors celui-ci n'aura pas à envoyer son ID serveur. Les ID serveurs sont crées par les serveur et routeurs eux-mêmes.

Puisque le protocole d'authentification de connexion SILC est toujorus exécuté après le protocole de SKE, les clés de session ont déjà été établies. cela signifie que tous les paquets envoyés au protocole d'authentification de connexion sont cryptés et authentifiés.

L'authentification peut reposer sur une phrase de passe ou une clé publique de cryptage. Il est aussi possible qu'elle ne nécessite pas du tout d'authentification. Un paquet envoyé par un client (par exemple) est totalement crypté, l'envoi de la phrase de passe à l'intérieur du paquet est donc sécurisé.

Si l'authentification repose sur une clé publique, alors, le client (par exemple) signe les données avec sa clé privée et les envoie au serveur. Le serveur vérifie alors la signature en utilisant la clé publique du client. Le paquet est aussi crypté dans le cas d'une authentification par clé publique. Si l'authentification échoue, la connexion au serveur ou au routeur sera refusée. Si elle marche, la connexion est autorisée. Après cela, le client est prêt à communiquer à l'intérieur du réseau SILC.

VII. Les canaux

Un canal est un groupe d'un client ou plus, avec un nom et qui recevra tous les messages adressés à ce canal. Il est créé lorsqu'un premier client s'y connecte et il cesse d'exister lorsque le dernier client le quitte. Lorsqu'un canal existe, n'importe quel client peut le rejoindre en utilisant le nom du canal. Un canal est un lieu où un groupe de gens peut engager une conversation.

Les noms de canal sont uniques sur le réseau SILC. Il ne peut pas y avoir simultanément deux canaux avec un nom identique. Cependant, un canal a quand même un ID de canal qui sert à référencer le canal sur le réseau. La taille maximum d'un nom de canal est de 256 caractères.

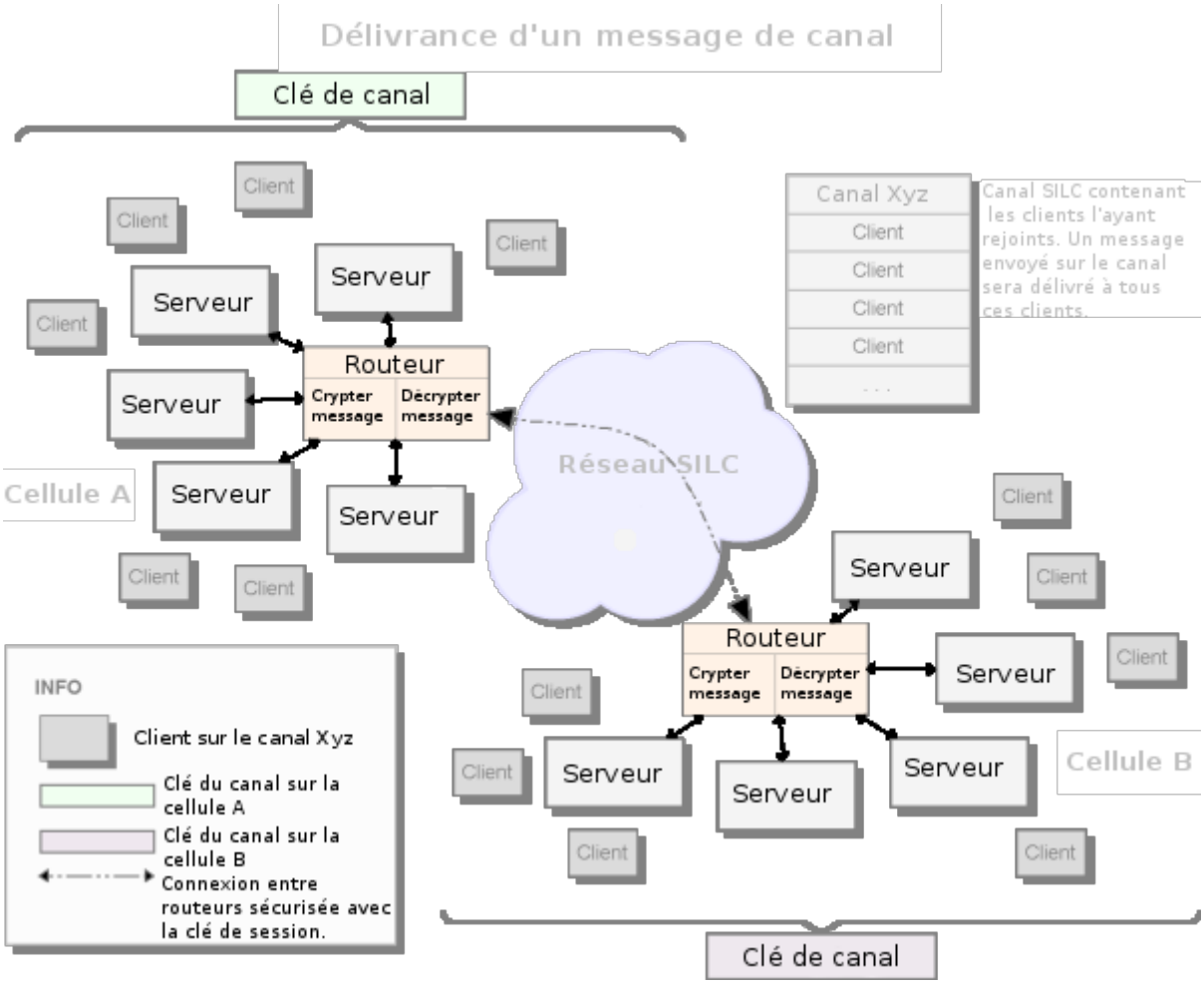
Les canaux peuvent avoir des opérateurs qui peuvent administrer ceux-ci et gérer tous ses modes. Il existe deux types d'opérateurs sur les canaux : le fondateur du canal et l'opérateur de canal. Le fondateur est le client qui a créé le canal, c'est aun opérateur avec quelques privilèges supplémentaires. Il peut gérer tous les modes du canal. De plus, les privilèges du fondateur ne peuvent être enlevés par aucun autre opérateur, et il ne peut pas être viré de force du canal. Il lui est aussi possible de regagner ses privilèges plus tard, et ce même s'il a quitté le canal.

L'opérateur de canal est un opérateur qui peut gérer la plupart des modes du canal et l'administrer. Cependant, il ne peut gérer tous les modes qui sont sont strictement réservés au fondateur. Ils peuvent néanmoins administrer le canal, régler certains modes, supprimer un client gênant, et promouvoir d'autres clients au rang d'opérateur du canal.

VII.1. La délivrance des messages de canal

Tous les clients ayant rejoint le canal peuvent envoyer des messages dessus. Tous les messages de canaux sont sécurisés et authentifiés par une clé de canal. Cette clé est générée par le serveur lors de la création du canal, à l'arrivée ou au départ d'un client. Elle est aussi regénérée périodiquement. La regénération de clé lors du départ ou de l'arrivée d'un client empêche aux nouveaux clients de lire les messages précédemment postés, et aux anciens de lire ceux qui viennent d'être postés. Ils peuvent uniquement crypter et décrypter les messages de canaux après avoir rejoint le canal.

Sur le réseau SILC, les clés de canal sont spécifiques à la cellule . Chaque cellule qui a des clients sur un canal en particulier possède sa propre clé pour celui-ci. Cette clé n'est pas partagée par d'autres cellules du réseau. A l'intérieur de la cellule, la clé de canal est connue par le routeur et tous les serveurs qui ont un client sur ce canal, ainsi que les clients ayant rejoint le canal.



Le diagramme ci-dessus illustre la délivrance typique d'un message de canal à l'intérieur d'une cellule, et entre deux cellules. Ces deux cellules possèdent leur propre clé de canal. Elles connaissent tous les clients présents sur le canal. Quand un message est envoyé sur le canal par un client, il est crypté avec la clé actuelle de la cellule. Les serveurs et le routeur de la cellule locale acheminent alors le message à tous les clients locaux qui sont présents sur le canal. Si le canal possède des utilisateurs appartenant à d'autres cellules du réseau, le routeur acheminera le message de canal à cette cellule. Quand les messages sont envoyés entre deux routeurs, ils sont d'abord décryptés avec la clé de canal actuelle, puis recryptés avec la clé de session partagée par les deux routeurs. Le routeur recevant le message de canal le décrypte alors avec la clé de session puis le recrypte avec la clé de canal en cours sur la cellule. Il distribue ensuite le message de canal à tous les clients du canal. Les clients présents sur le canal savent toujours la clé de canal en cours et peuvent donc décrypter tous les messages de canal qu'ils recoivent. Notez que les serveurs normaux sur le réseau SILC ne décryptent jamais les messages de canal même s'ils en possèdent la clé. Les serveurs n'ont aucune raison de décrypter le message. Les routeurs décryptent le message seulement quand ils doivent l'envoyer à un autre routeur.

Cette méthode de délivrance du message de canal est par défaut celle qui envoie les messages de canal sur le réseau SILC. Cependant, ce n'est pas une solution appropriées dans toues les circonstances. Si les clients présents sur un canal précis ne peuvent faire confiance, ou ne veulent pas faire confiance aux serveurs et aux routeurs du réseau SILC, ils peuvent considérer que la connaissance de la clé de canal par les routeurs et serveurs est une faille de sécurité.

D'un autre côté, si les clients peuvent faire confiance aux serveurs et routeurs du réseau SILC, il

s'agit alors de la meilleure façon d'envoyer des messages de canal. Cette méthode est la plus simple pour l'utilisateur final puisqu'elle ne requiert aucune options spéciales avant d'engager la conversation sur le canal. Le client rejoint simplement le canal, reçoit la clé de canal de la part du serveur et peut commencer la conversation.

En plus de crypter les messages de canal, il est possible de signer numériquement tous les messages de canal envoyés. Le récepteur peut alors vérifier la signature de chaque message en utilisant la clé publique de l'expéditeur.

VII.2. La délivrance des messages de canal en utilisant une clé privée de canal

Si les clients ne peuvent pas faire confiance aux serveurs et aux routeurs du réseau, ils ne devraient pas utiliser la méthode par défaut d'envoyer des messages de canal. Ils devraient plutôt se servir des clés privées de canal pour crypter et décrypter les messages de canal. Ces clés privées sont des clés qui sont connues seulement par les clients qui sont présents sur le canal. Les serveurs et les routeurs ne connaissent pas cette clé, et ne peuvent donc pas décrypter les messages. Lorsqu'un message est envoyé entre deux routeurs il est simplement recrypté avec la clé de session mais il n'est pas décrypté puisque le routeur n'a pas la clé qu'il faut pour le faire.

Les clients présents sur le canal doivent d'abord accepter la clé privée de canal qu'ils vont utiliser. Il peut généralement s'agir de n'importe quoi. Ce peut-être une phrase de passe, une chaîne de caractères aléatoire, ou bien la clé peut être négociée en usant d'un protocole de SKE qui fournit simultanément et à plusieurs clients la négociation de clé.

Puisque la clé privée de canal est un réglage effectué par le client, il est possible de mettre plusieurs clés privées de canal pour un seul canal. Il est aussi possible d'avoir de multiples clés privées de canal qui ne sont pas connues par tous les membres. Lors du cryptage des messages avec une clé privée de canal, seul les clients possédant cette clé peuvent décrypter ce message. Une autre clé peut être partagée par tous les clients du canal qui pourront alors décrypter tous les messages cryptés avec celle-ci. De cette façon, il devient alors possible d'avoir un groupe de discussion privé à l'intérieur d'un canal tout en participant à la conversation générale.

VIII. Messages privés

Les messages privés sont des messages envoyés d'un client à un autre à travers le réseau SILC. Ils sont privés car ils ne sont envoyés à personne d'autre que le vrai destinataire du message. Ces messages privés peuvent être utilisés afin d'engager une discussion privée avec un autre client, si celle-ci n'est pas envisageable sur le canal.

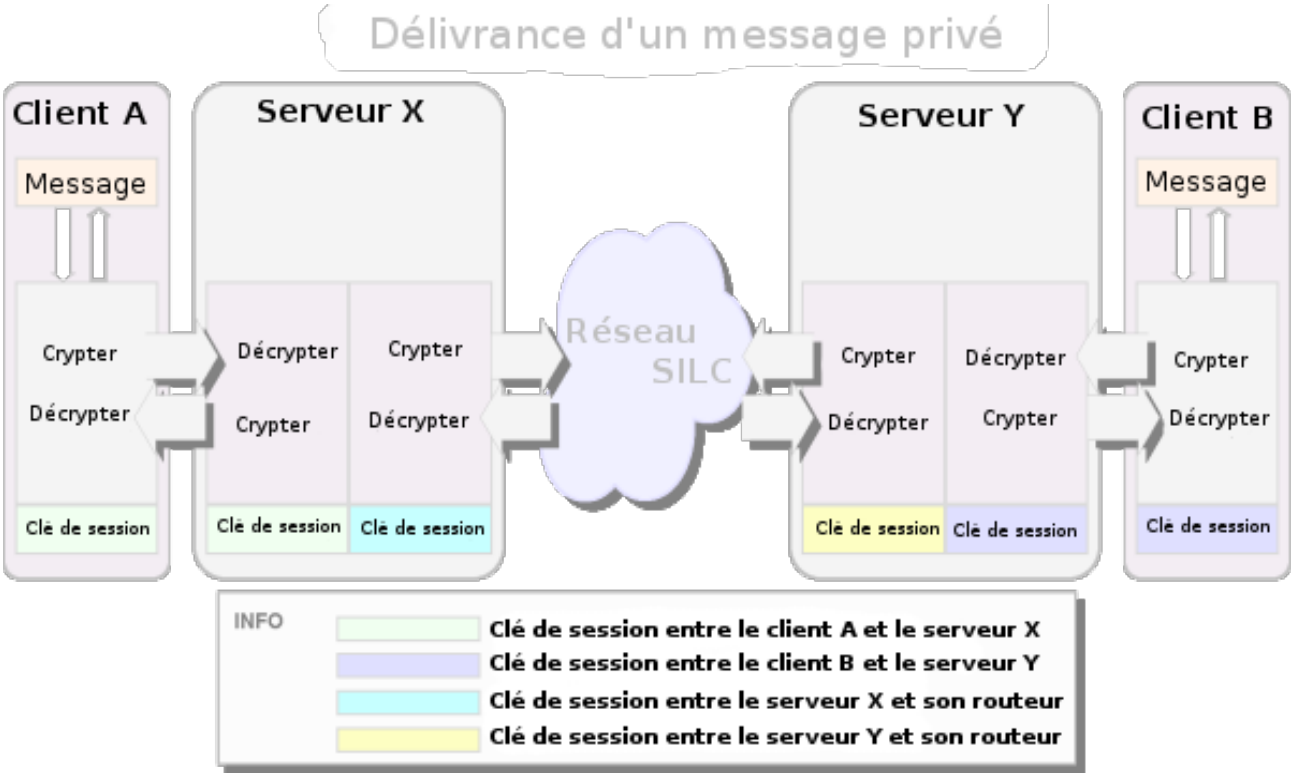
Comme tous les messages de SILC, le message privé est lui-aussi crypté et authentifié. Il existe un grand nombre de manières de sécuriser des messages privés. Par défaut, ils sont cryptés avec les clés de session établies lors du protocole de SKE. Il est aussi possible d'échanger une clé de message privé enter les deux clients, et de crypter les messages avec cette clé. Il est même possible de crypter les messages avec un cryptosystème à clé publique. la prochaine section décrira toutes ces méthodes de délivrance des messages.

Le protocole SILC fournit ces trois méthodes de délivrance des messages privés car aucune d'entre elle ne peut, à elle seule, satisfaire les exigences de sécurités de tout le monde. L'utilisateur final doit avoir le choix du niveau de risque acceptable, du niveau de sécurité nécessaire et d'autres aspects de sécurité et d'usage. Et c'est pour cette raison que l'utilisateur a le choix entre plusieurs méthodes d'envoi des messages privés.

En plus de crypter les messages privés, il est aussi possible de les signer numériquement. Le destinataire pourra ainsi vérifier la signature de chacun des messages en utilisant la clé publique de l'expéditeur.

VIII.1. La délivrance des message privés utilisant les clés de session

L'envoi de messages privé est sécurisé par défaut avec les clés de session établies durant le protocole de SKE. Cela signifie que les messages privés sont toujours cryptés avec la clé de session du prochain récepteur du message en cours d'acheminement vers le client destinataire. Cela veut aussi dire que le message est décrypté et recrypté chaque fois qu'il est renvoyé jusqu'à ce qu'il atteigne le client destinataire.



Comme le montre le diagramme ci-dessus, les messages privés envoyés par le client A vers le client B voyagent à travers le réseau SILC et sont toujours décryptés et recryptés avec la clé de session du prochain récepteur du message. Le client B décrypte finalement les messages privés qui sont cryptés avec la clé de session partagée par le client B et le serveur Y.

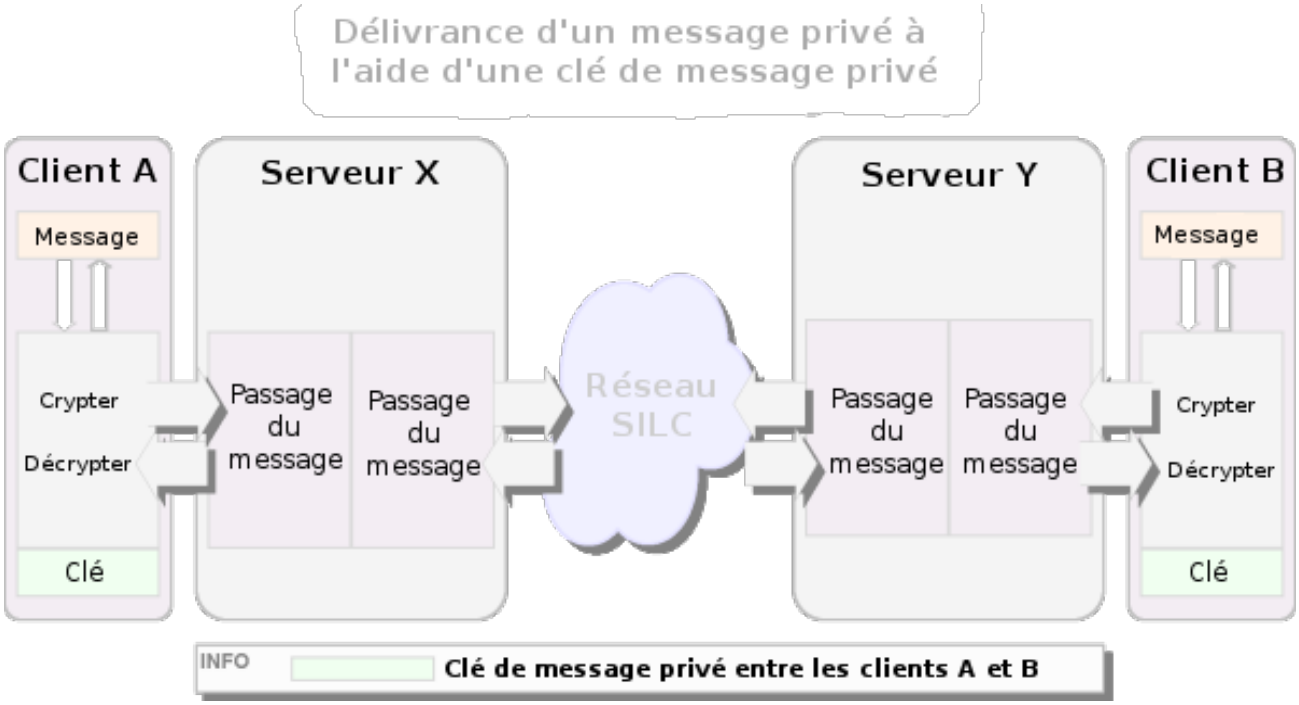
Cette méthode de sécurisation des messages privés n''est pas parfaite et ne peut pas être employée dans toutes les circonstances. Si les clients discutant ne peuvent pas faire confiance aux serveurs et aux routeurs du réseau SILC, ils ne devraient pas envoyer de messages privés sécurisés ainsi. Les messages sécurisés de cette manière peuvent en effet être décryptés par des serveurs et des routeurs auxquels le client ne fait pas confiance.

D'un autre côté, si les clients peuvent faire confiance aux serveurs et routeurs du réseau SILC, ou

qu'ils se moquent que les serveurs puissent décrypter leur messages, alors l'envoi de messages privés par cette méthode est la plus simple du point de vue du client. Cela implique, bien entendu, que les serveurs et les routeurs décryptent et recryptent chaque message privé. Puisque cette méthode n'est pas toujours employable, le protocole SILC fournit d'autre méthode de sécurisation des messages privés.

VIII.2. La délivrance des messages privés utilisant une clé de message privé

Les messages privés peuvent de même être sécurisés avec une clé de message privée. Cette clé n'est connue que de l'expéditeur et du destinataire du message . Ainsi, personne d'autre ne peut crypter et décrypter le message. Celui-ci est crypté par l'expéditeur avec la clé de message privé et tous les serveurs et routeurs transmettent le message juqu'au destinataire. Ils ne peuvent décrypter le message puisqu'ils n'en possèdent pas la clé. Lors de l'envoi de messages privés par cette méthode, le fait que les clients fassent confiance ou pas aux routeurs et serveurs du réseau SILC n'importent pas.



Comme l'illustre le diagramme ci-dessus, le client A crypte le message avec la clé de message privé puis l'envoie sur le réseau SILC. Tous les serveurs et routeurs transmettent simplement le message puisqu'ils ne peuvent pas le décrypter. Le client B reçoit le message et le décrypte avec la clé de message privé.

L'envoi de messages privés de cette façon est toujours sécurisé puisque la clé est partagée seulement par l'expéditeur et le récepteur. Le problème de cette méthode repose sur la manière dont l'expéditeur et le destinataire vont se mettre d'accord sur la clé à utiliser. La clé de message privé peut être n'importe quoi, comme une phrase de passe par exemple. Ils peuvent s'être entendus au préalable, c'est-à-dire avant le début de la discussion, sur un mot ou une phrase à utiliser en tant que clé. La clé peut aussi être une chaîne aléatoire issue d'un livre qu'eux seuls possèdent. Il peut encore

s'agir d'une clé échangée en se servant d'un protocole de SKE.

Le problème est néanmoins fondamental. Comment se mettre d'accord sur la clé à employer quand vous êtes dans l'incapacité de joindre la personne sur un canal sécurisé ? Le protocole SILC résoud ce problème en fournissant une possibilité d'échange de clé entre deux clients par le biais d'un protocole de SKE. L'un ou bien les deux clients peuvent mettre en place un serveur de SKE sur leur machine et demander à l'autre client de s'y connecter. Dans ce cas le SKE est exécuté en dehors du réseau SILC. Comme résultat du protocole de SKE, les deux clients partagent maintenant un secret qu'ils puevent utiliser comme clé de mmessage privé. La clé est uniquement connue par les deux clients qui ont lancé le protocole de SKE. Ils peuvent utiliser cette clé pour échanger tous les messages privés ultérieurs.

L'utilisation de cette méthode de délivrance des messages privés est recommandé dans le cas où les clients ne peuvent faire confiance aux routeurs et aux serveurs du réseau SILC. L'inconvénient est la phase supplémentaire de mise en place de la clé de message privé avant le début de la discussion. Cependant, l'utilisation du protocole de SKE est la méthode recommandée pour échanger la clé puisqu'elle peut être automatisée et ne nécessitent aucune action supplémentaire de la part de l'utilisateur final.

IX. Les messages MIME

Le protocole SILC supporte les messages MIME sur les canaux normaux et ceux de messages privés. Les messages MIME permettent d'envoyer des images, des flux audios et vidéos et de la musique sur SILC. Tous les types MIME qui sont supportés par l'application peuvent être envoyés via le réseau SILC.

Les messages MIME reposent sur l'emploi de marqueurs de message (Message Flags) dans la charge utile des messages utilisés dans le protocole SILC. Les marqueurs de message indiquent au destinataire du message que celui-ci est de type MIME, il sait alors comment l'interpréter. L'utilisation de ces marqueurs permet de même l'envoi d'autre sorte de messages et l'augmentation des fonctionnalités des canaux normaux et des canaux de messages privés.

X. Les transferts de fichier sécurisés

Le support du transfert de fichier sur les protocoles de chat est aujourd'hui une obligation, sans cela, on ne peut même pas le considérer comme un protocole de chat. SILC supporte le transfert de fichier, mais surtout ce flux de transfert de fichier est sécurisé. lorsqu'un utilisateur désire transférer un fichier à un autre utilisateur, le protocole de SKE est d'abord exécuté pour échanger une clé de session pour le flux de transfert de fichier. Cette clé est alors employée pour protéger le flux de pair à pair entre utilisateurs.

Le protocole de transfert de fichier employé par le protocole SILC est le protocole de transfert de fichier par SSH (SSH File Transfer protocol - SFTP). Bien que le nom du protocole fasse mention de SSH, le protocole de transfert de fichier n'a rien à voir avec le Shell Sécurisé (Secure Shell - SSH). Le SFTP est un protocole totalement indépendant et son flux est sécurisé en utilisant SILC. Le SFTP est un très bon protocole car, en plus de fournir le support de simple transfert de fichier, il peut aussi supporter des manipulations complexes de fichiers et dossiers.

Le support du transfert de fichier sur SILC a été conçu afin d'être utilisé avec n'importe quel protocole de transfert de fichier. Le protocole obligatoire est SFTP mais dans le futur, le support d'autres protocoles est à rpévoir.

XI. Le futur du protocole

Le protocole a mûri jusqu'à atteindre la version 1.2 ces dernières années. Il a atteint un niveau tel qu'il est le protocole de discussion le plus fonctionnel à ce jour. L'intention du projet SILC est de standardiser le protocole SILC à l'IETF, et c'est ce vers quoi notre regard se tourne en ce moment.

XII. Conclusion

Discussion en directe sécurisée sur internet (Secure Internet Live Conferencing - SILC) est un protocole moderne de discussion qui fournit un grand nombre de fonctionnalités de discussion avec une haute sécurité. Il possède une vaste gamme de propriétés de sécurité et de fonctionnalités qui répondent aux hautes exigences de sécurité, tout en restant facile d'usage. La topologie du réseau offre une nouvelle solution d'architecture avec une meilleure évolutivité que les protocoles de chat traditionnels.

XIII. Pour aller plus loin

De plus amples informations à propos du protocole SILC sont disponible dans les documents de spécification du protocole SILC. Il existe actuellement six brouillons Internet (Internet Drafts) qui définissent le protocole dans tous ses détails. Ces brouillons sont disponibles sur le site web du projet SILC (http://silcnet.org/) mais aussi sur le site de l'IETF (http://www.ietf.org/). Pour une introduction compréhensible à la cryptographie, veuillez vous référer au document "Cryptography A-2-Z" (http://www.ssh.com/tech/crypto/).

XIV. Termes & Abbréviations

- Cryptosystème asymétrique

Un cryptosystème asymétrique fournit un cryptage public. Il possède deux clés, une clé publique et une clé privée (parfois appelée clé secrète). La clé publique est disponible publiquement, permettant ainsi à tout le monde de crypter des messages avec celle-ci. Seul le possesseur de la clé privée est à même de décrypter ces messages. La différence avec les cryptosystèmes symétriques est que ces derniers n'utilisent qu'une seule clé, et celle-ci sert au cryptage et au décryptage. Le cryptosystème asymétrique est aussi appelé cryptage à clé publique, cryptosystème à clé publique ou algorithme à clé publique. SILC supporte les cryptosystèmes asymétriques RSA et DSS.

- Authentification

La vérification de l'identité d'une personne, d'un hôte ou d'un processus dans le but d'obtenir l'accès à un service, ou de prouver son identité. Dans les communications de données, cela signifie aussi la vérification de l'origine d'un message.

- Certificat

Un certificat est un document numérique qui peut être employé afin de vérifier l'identité d'une personne ou d'un hôte. Sur SILC, les certificats peuvent être utilisés pour prouver l'identité de clients, de serveurs et de routeurs. En gros, un certificat est une clé publique avec un nom de sujet. SILC supporte les certificats X.509, OpenPGP et SPKI. Les clés publiques supportés sont les clés publiques de type SILC et celles de type SSH2.

- Autorité de certification (Certification Authority - CA)

Une tierce partie qui peut vérifier l'identité d'une personne ou d'un hôte. La CA est généralement une entreprise extérieure qui fournit des certificats et des services pour les vérifier.

- Echange de clé de Diffie-Hellman

Le premier algorithme à clé publique inventé. Il sert à générer une clé secrète entre deux parties ou plus. Il tire sa sécurité de la difficulté à calculer des logarithmes discrets.

- Cryptage

Un mécanisme (habituellement matématiques) pour passer d'un texte en clair à un texte chiffré afin de fournir de la confidentialité. Un procédé de passage du texte chiffré vers le texte en clair est appelé décryptage.

- Intégrité

La vérification de données afin de détecter n'importe quelles modifications. Si les donnée sont modifiés sur le chemin de l'expéditeur au destinataire, la modification sera détectée.

- HMAC

Hashage du code d'authentification de message (Hash Message Authentication Code), aussi appelé fonction de hachage à clé. C'est un algorithme d'authentification de clé secrète qui prouve que le message n'a pas été modifié et que le HMAC a été calculé par l'expéditeur du message.

- Gestion de clé

La gestion de clé est un ensemble de processus et de mécanismes qui supporte l'échange de clé et le maintien à jour des clés entre les différentes parties, cela inclue le remplacement des vieilles clés par de nouvelles (si nécessaire), par le lancement du reclé.

- Attaque par l'homme-du-milieu (Man-in-the-middle attack)

Une attaque contre deux entités connectés entre elles, où l'attaquant lance le protocole d'échange de clé avec chacune des deux parties, et sans qu'elles le sachent. Les deux entités connectées finiront par se servir de la clé secrète définie par l'attaquant, et celui-ci pourra alors crypter et décrypter tous les messages entre les deux entités.

- Code d'authentification du message (Message Authentication Code - MAC)

Un MAC assure l'intégrité du message. Il est calculé en utilisant un algorithme d'authentification à clé secrète (HMAC).

- Secret ultérieur idéal (Perfect Forward Secrecy - PFS)

Une propriété du reclé (ou régénération de clé) qui définit si la nouvelle clé dérive ou pas de l'ancienne. Si cette option est active, la nouvelle clé ne dépendra jamais de l'ancienne, ainsi, l'ancienne clé devait être compromise plus tard, cela ne compromettrait pas la nouvellé clé. Dans SILC, l'activation du PFS dans le protocole de SKE relancera le protocole de SKE. Si le PFS n'est pas sélectionné, la nouvelle clé dérivera toujours de l'ancienne.

- Reclé

Un processus de régénération de clé utilisé lorsque la clé a expiré ou qu'elle n'est plus sûre. Dans ce cas, le reclé est lancé et une nouvelle clé est générée.

- Cryptosystème symétrique

Un cryptosystème symétrique est un cryptosystème où une clé unique est utilisée pour les processus de cryptage et de décryptage. Les cryptosystèmes symétriques sont habituellement bien plus rapides que les cryptosystèmes asymétriques. DES, AES, Twofish et blowfish en sont des exemples. SILC supporte tous les cryptosystèmes symétriques y compris AES. SILC ne supporte pas DES car il n'est pas sécurisé et 3DES car il est trop lent.