

SILC Protocol White Paper

Pekka Riikonen - SILC Project

Version 1.2 / 22 October 2003

SILC Protocol White Paper

Version 1.2 / 22 October 2003

1 Introduction

Chat protocols are very popular on the Internet. They have actually been very popular since the very first chat protocols appeared on the net. The Internet Relay Chat (IRC) was one of the first chat protocols, and quickly gained the status of being the most popular chat on the net. Today, IRC has several competitors from various other so called Instant Messaging (IM) protocols, such as ICQ. However, all of these different chat protocols have something in common; they are all insecure.

The security is important feature in applications and protocols in contemporary network environment. The older chat protocols, however have failed to meet the growing security requirements on the Internet. It is not anymore enough to just provide services, like for example chat services. Now, they need to be secure services.

The Secure Internet Live Conferencing (SILC) protocol is a new generation chat protocol which provides full featured conferencing services, just like any other contemporary chat protocol provides. In addition, it provides security by encrypting and authenticating the messages in the network. The security has been the primary goal of the SILC protocol and the protocol has been designed from the day one security in mind. All packets and messages travelling in the SILC Network are always encrypted and authenticated. The network topology is also different from for example IRC network. The SILC network topology attempts to be more powerful and scalable than the IRC network. The basic purpose of the SILC protocol is to provide secure conferencing services.

The SILC Protocol have been developed as Open Source project. The protocol specifications are freely available and they have been submitted to the IETF. The protocol is currently stabilizing and has reached a version 1.2.

2 About This White Paper

The purpose of this white paper is to give short but deep enough introduction to the SILC Protocol. The document describes the purpose of the protocol and how the protocol works in practice. This document is intended for all audience. This document should be easy to understand for non-technical person and still be detailed enough for technically oriented person. See the section Terms and Abbreviations for terms used in this document.

(c) Copyright 2001 - 2003 Pekka Riikonen (priikone at silcnet.org)

This document is free document; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

3 SILC Protocol

Secure Internet Live Conferencing, or SILC in short, is a modern conferencing protocol which provides rich conferencing features with high security. One of the main design principles of the protocol was security. Many of the SILC features are found in traditional chat protocols such as IRC but many of the SILC features can also be found in Instant Message (IM) style protocols.

SILC combines features from both of these chat protocol styles, and can be implemented as either IRC-like system or IM-like system. In fact, SILC removes the need to make such distinction between these two protocol styles. Some of the more advanced and security features of the protocol are new to all conferencing protocols. SILC also supports multimedia messages and can also be implemented as a video and audio conferencing system. The protocol is also compact and robust and suites well for mobile environments where the low bandwidth sets special requirements for protocols. All packet sizes in SILC can be even further reduced by utilizing compression.

The packets and messages in the SILC network are always encrypted and authenticated. It is not possible to send unencrypted messages in SILC at all. This assures that end user cannot even accidentally send unencrypted messages while thinking that it is encrypted. This is one of the problems of most of the other chat protocols that provide so called plugin encryption. They are not secure by default but try to provide security by applying external security protocol such as PGP or SSL over the insecure protocol. In these cases the security is achieved usually by encrypting the data while key management, message authentication and other security issues may be left out, leaving the implementation vulnerable to various security problems. The other problem is also that the external protocols tend to leave the network only partly secured; usually only two points in the network are secured with for example SSL. While SSL does provide provable security it is not enough to provide security for a chat network as a whole.

SILC is secure in environment of mutual distrust between entities in the network. It is possible to encrypt messages end to end, so that only the sender and the receiver is able to encrypt and decrypt messages. It is also possible to send messages to group of users, so that only the specified group of users is able to encrypt and decrypt messages. Many times the protocol use keys that are generated by the servers, so that if other external key exchange methods fail the network still remains encrypted. However, it is always possible to negotiate and use locally generated keys to secure messages, so that the servers do not know the key.

Like so many other contemporary chat protocols, SILC too provides file transfer. It is possible to transfer files securely between users in the SILC Network. The actual file transfer stream is always sent outside the network peer to peer. Before the file transfer is started a key exchange protocol is executed to negotiate file transfer session key.

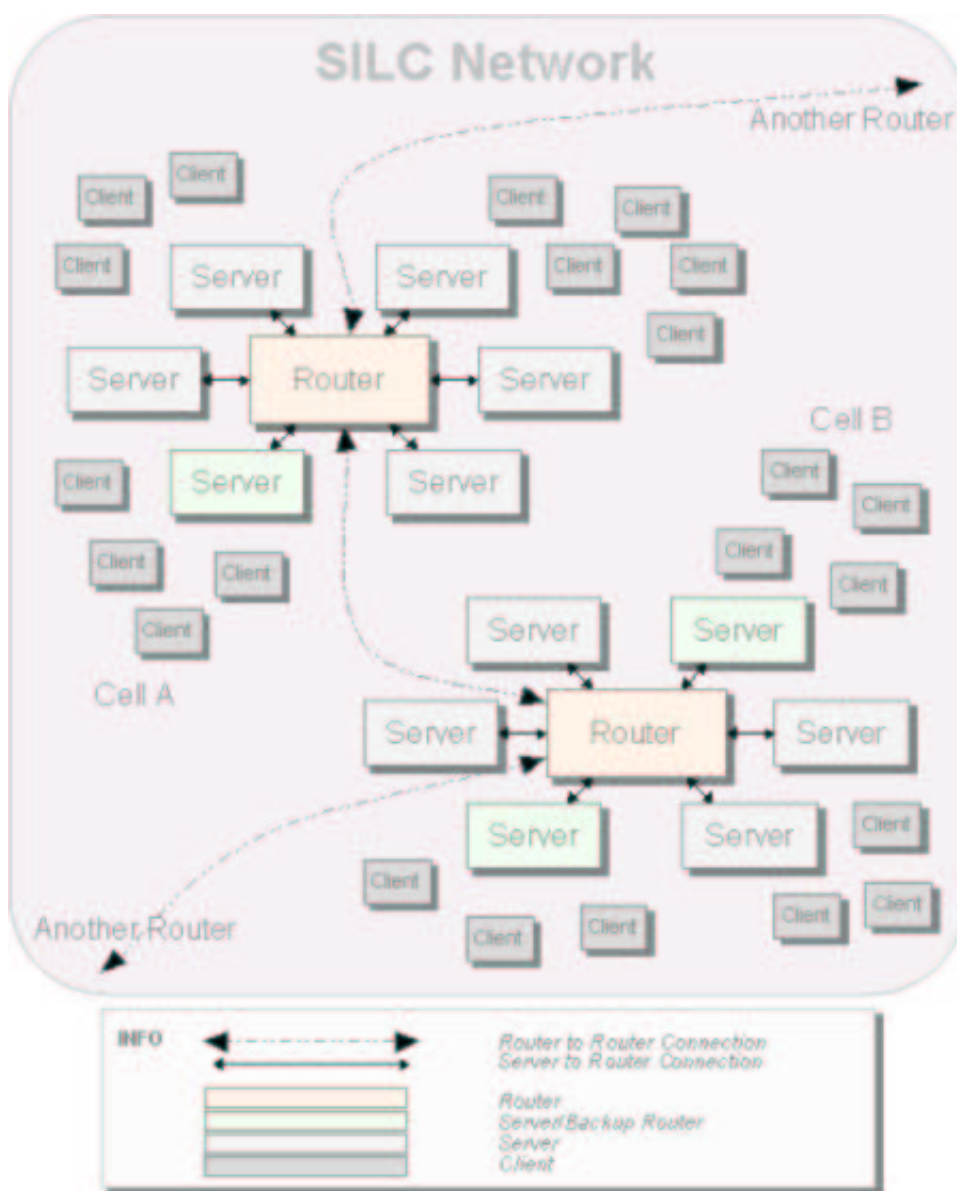
The SILC protocol also supports so called detaching, a novel idea where it is possible to detach from the server without actually quitting the network. It is then later possible to resume the connection back to some server in the network, and be like you were never gone.

The SILC protocol also allows distribution and exchange of public keys and certificates through the SILC network. It is also possible to fetch detailed user information from other users through the SILC network. It is possible to fetch for example users's business card,

pictures, certificates, etc. SILC protocol also supports secure file transfer to allow document and file exchange securely between users.

SILC protocol also supports services, which are extensions to the core protocol. They can be used to augment the features of the protocol or to add entirely new features without breaking backwards compatibility. Services can be negotiated online and authenticated with passphrases or with digital signatures.

The network topology is also different from traditional conferencing and chat protocols. The SILC network forms so called hybrid ring-mesh network at the router level, and star network at the server level. This sort of network topology allows better scalability and faster delivery of packets than traditional spanning tree style network. The router servers and normal servers also has the distinction that only router's know global information and keep the global network state up to date, and normal servers keep only local information up to date. This significantly increases the scalability of the network. The network also supports backup routers which can be used to protect the network against netsplits.



The diagram above illustrates a portion of the SILC network. It shows two cells that both have several servers, and backup routers and several clients. Clients can connect to server and routers if they want to. The following sections will describe the entities of the SILC Network in greater detail.

3.1 Clients

A client is a piece of software connecting to SILC server. The software is usually run by the end user, a real person that is. The purpose of the clients is to provide the end user an interface to the SILC services. They are used to actually engage the conversations on the SILC Network, and they can be used to execute various SILC commands.

The clients are distinguished from other clients by unique Client ID. There cannot be multiple same Client IDs in the SILC Network at the same time. The end user, however does not use Client IDs. The end users usually selects a preferred nickname they want to use, and identifies themselves with that nickname to other users on the network. The nicknames are not unique in the SILC Network. There can be multiple same nicknames at the same time on the network. The maximum length for the nickname is 128 bytes.

Most of the other chat protocols have unique nicknames. This is where SILC differs from most of the other chat protocols. The purpose of this feature is to make IRC style nickname wars obsolete, as no one owns their nickname; there can always be someone else with the same nickname. This feature also makes nickname registering services obsolete.

When client connects to the server the SILC Key Exchange (SKE) protocol and SILC Connection Authentication protocol are executed. The result of the SKE protocol is the session key that the client and server use to secure their communication. All commands, for example, that the client sends to the server are secured with the session key. The session key expires periodically and the rekey process can be executed with or without the Perfect Forward Secrecy (PFS). The connection authentication protocol is used to authenticate the client to the server. The server may allow the client to connect without authentication, or it may require a passphrase or public key based (or certificates) authentication.

3.2 Servers

Servers forms the basis for the SILC Network, by providing a point to which clients may connect. There are two kinds of servers in SILC; normal servers and router servers. The next section describes the function of router server.

Normal servers connect to router server. Normal servers cannot directly connect to other normal servers. Messages that are destined outside the local server are always sent to the router for further routing. The clients usually connect to the normal server, however, clients may connect to router servers as well. The SILC Network diagram above illustrates how normal servers connects to the router server.

The servers are distinguished by other servers in the network by unique Server ID. There cannot be multiple same Server IDs in the SILC Network at the same time. The servers keep track of local information. It knows all locally connected clients and it knows all channels that its clients have joined. However, it does not know any global information. It usually does not keep track of global clients, however, it may cache that information if it was queried. The reason for this is that the server does not need to keep global information up to date and thus

makes the server faster (and in the end the entire network faster). They can always query the information from the router.

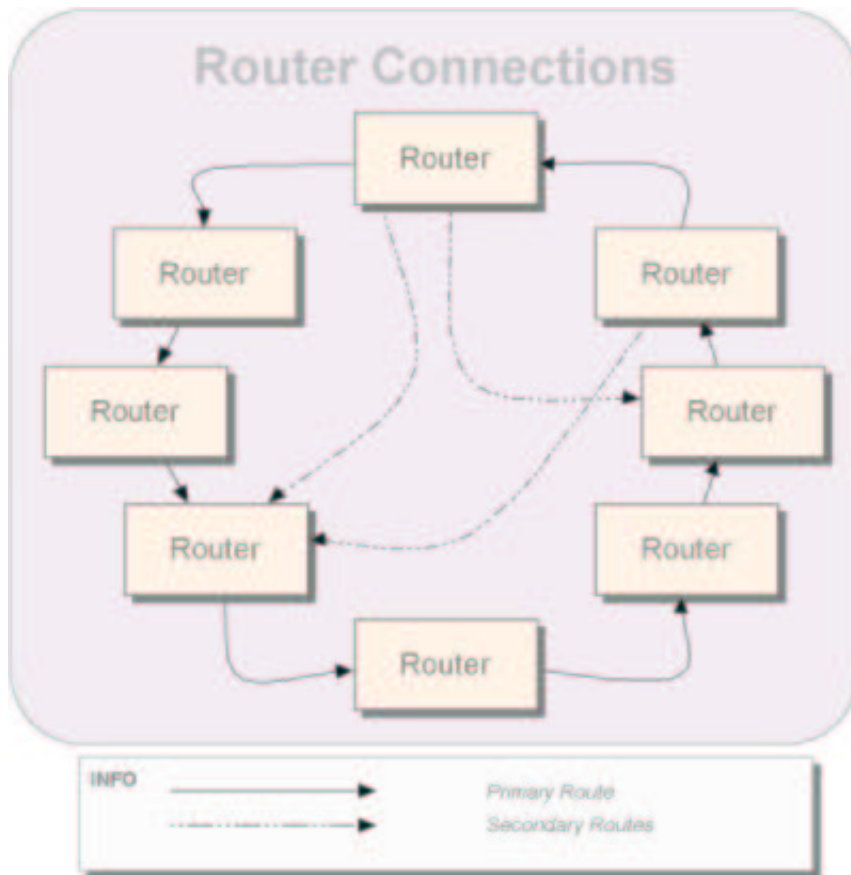
When server connects to its router the SILC Key Exchange (SKE) protocol and the SILC Connection Authentication protocol are executed, just like when client connects to server. The SKE results in to the session key that is used to secure the communication between the server and the router. The connection authentication protocol is used to authenticate the server to the router. The authentication is always based in either passphrase or public key (or certificates).

3.3 Routers

The router servers are servers that actually handles the message routing in the network. They are, however also normal servers and they do accept client connections. Each of the router in the network is called a cell. A cell can have only one active router and it may have several servers and several clients. The cell, however may have backup routers that can take over the tasks of the primary router if it becomes unresponsive. The switch to the backup router should be transparent and only local connections to the primary router are lost. Other connections in the cell are intact, and clients and servers merely experience some lag in the network connection during the switch to the backup router.

The normal server knows only local information. Router server on the other hand knows local information and global information. It considers the cell as local and outside cells as global. It knows all the clients connected to the network, all created channels, and all routers and servers in the network. The server may query the global information if it is needed. For example, when client sends WHOIS command, the server may query the information from the router. If the router does not know all the details that the WHOIS command requires it can query the information from a router or a server which knows all the details. It may then cache that information.

The primary purpose of the router server is to route the messages to local servers and local clients, and messages that are destined to outside the cell are routed to the primary route or some other secondary route if it is a faster route. The routers in the network forms a ring. Each router has a primary route to other router in the network. Finally the ring is closed by the last router using the first router in the network as its primary route.



The diagram above illustrates how the routers form a ring in the network. A router may have several secondary routes which it may use when it routes the packets.

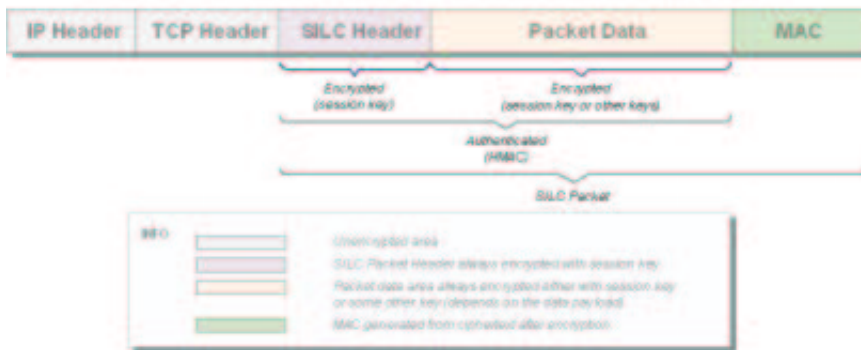
When routers connect to its primary router the SKE and the SILC Connection Authentication protocols are executed just like when normal server connects to its router. The session key is used to secure the communication between the routers. All the secondary routes also have their own session keys.

4 SILC Packet Protocol

The basis of SILC protocol relies in the SILC packets and they are without a doubt the most important part of the protocol. The SILC Packet protocol is a secure binary packet protocol. The protocol provides secure binary packets and assures that the contents of the packets are secured and authenticated.

Packets are used in the SILC protocol all the time to send for example channel messages, private messages, commands and other information. All packets in SILC network are always encrypted and their integrity is assured by computed Message Authentication Codes (MAC). The protocol defines several packet types and packet payloads. Each packet type usually has a specific packet payload that actually defines the contents of the packet. Hence, the actual data in the packet is the packet payload defined in the protocol.

SILC Packet and Packet Encryption



As the diagram above illustrates the SILC packet is constructed from the SILC Packet Header that is included in all SILC packets, data area that includes the packet payloads, and MAC area which assures the integrity of the packet. Entire SILC packet is always encrypted, except for the MAC area which is never encrypted. The encryption process and the key used, however depends on the packet payload. Some of the payloads are encrypted with the session key and some are encrypted with other keys, for example with channel message keys. The SILC Packet Header is always encrypted with the session key. The MAC is computed from the SILC Packet Header and the data area after encryption. This is so called Encrypt-Then-MAC order.

5 SILC Key Exchange Protocol

SILC Key Exchange Protocol (SKE) is used to exchange shared secret between connecting entities. The result of this protocol is a key material used to secure the communication channel. This protocol is executed when, for example client connects to server. It is also executed when server connects to router. And, there is no reason why it could not be executed between two clients too, if two clients would need to create secret key. The purpose of the SKE protocol is to create session keys to be used in current SILC session. The SKE is based on the Diffie-Hellman key exchange algorithm, and is immune to for example man-in-the-middle attacks by using digital signatures.

This is the first protocol that is executed when creating connection to, for example SILC server. All the other protocols are always executed after this protocol. This way all the other protocols are secured since the SKE creates the session key that is used to secure all subsequent packets. The session keys created in the SKE are valid only for some period of time (usually an hour) or at most until the session ends. The rekey process can be executed with or without the Perfect Forward Secrecy (PFS).

The security properties that are used in the SILC session are also negotiated during the SKE. The protocol has initiator and responder. The initiator is the one who starts the SKE negotiation and responder is the one who receives the SKE negotiation. When the protocol is started initiator sends a list of security properties that it supports. The responder then selects the security properties it supports and sends its reply to the initiator. The security properties includes ciphers, hash functions, public key algorithms, HMAC functions and other security properties. The responder can always choose the properties it supports.

After the security properties are selected the protocol continues by performing the Diffie-Hellman key exchange algorithm. At the same time the initiator and responder also sends their public keys or certificates to each other. The initiator and responder also computes a signature that the other party will verify. By default the protocol is executed in so called mutual authentication mode, where both of the parties computes a signature which are verified by each other independently. This way both of the parties will have prove the possession of the private key to the public key they are providing in the protocol. If any of the phases of the protocol are to fail the connection is closed immediately.

The public key or certificate that is received during the SKE protocol must be verified. If it is not verified it would be possible to execute a man-in-the-middle attack against the SKE protocol. If certificates are used they can be verified by a third party Certification Authority (CA). Verifying a public key requires either confirming a fingerprint of the public key over phone or email, or the server can for example publish the fingerprint (and the public key) on some website. In real life systems accepting the public key without verification, however is often desired. In many security protocols, such as in SSH2, the public key is accepted without verification in the first time when the connection is created. The public key is then cached on local hard disk. When connecting next time to the server the public key on local cache is verified against the public key server sent. In real life this works most of the time. However, if client (or server) cannot trust this, it must find some other way to verify the received public key or certificate.

6 SILC Connection Authentication Protocol

Purpose of SILC Connection Authentication protocol is to authenticate the connecting party with server or router. This protocol is executed when for example client connects to server. It is also executed when server connects to router. Its other purpose is to provide information for the server about which type of connection it is. The type of the connection defines whether it is client, server or router. If it is client then the server will create a new Client ID for the client. If it is server then it will expect the server to send its Server ID. Server IDs are created by the servers and routers itself.

Since the SILC Connection Authentication protocol is always executed after the SKE protocol, session keys has been established already. This means that all packets sent in the connection authentication protocol are encrypted and authenticated.

The authentication may be based either in passphrase or public key encryption. It is also possible to not require authentication at all. If the authentication is based to passphrase the passphrase is sent to the server. As the packet sent by, for example client, is entirely encrypted it is safe to send the passphrase inside the packet.

If the authentication is based to public key then, for example the client, signs data with its private key and sends it to the server. The server then verifies this signature by using the client's public key. The packet is also encrypted in the case of public key authentication.

If the authentication is to fail the connection to the server or router will be refused. If it is successful the connection is granted. After this the client is ready to communicate in the SILC Network.

7 Channels

A channel is a named group of one or more clients which will all receive messages addressed to that channel. The channel is created when first client joins to it, and the channel ceases to exist when the last client leaves it. When channel exists, any client can reference it using the name of the channel. Channel is a place where group of people can engage conversation.

Channel names are unique in the SILC Network. There cannot be multiple same channels in the network at the same time. However, channel has also a Channel ID which is actually used to reference the channel in the SILC Network. The maximum length for the channel name is 256 characters.

Channels can have operators that can administrate the channel and operate all of its modes. There are two types of operators on the channel: channel founder and channel operator.

The channel founder is the client which created the channel. Channel founder is channel operator with some more privileges. Channel founder can operate all of the channel's modes. Furthermore, channel founder privileges cannot be removed by any other operator on channel and channel founder cannot be removed from the channel by force. It is also possible for the channel founder to regain its privileges at later time, even if they have left the channel.

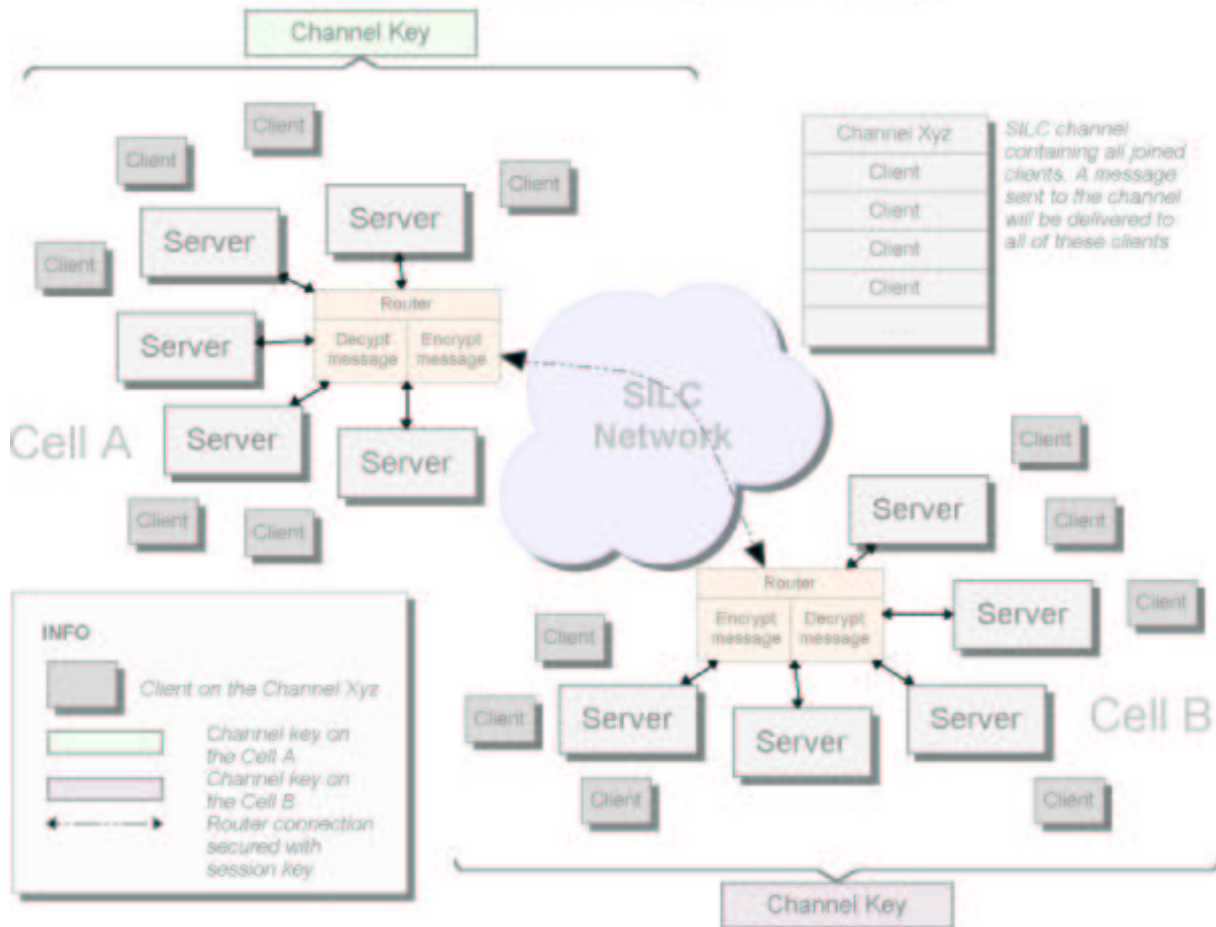
Channel operator is operator that can operate most of the channel's modes and administrate the channel. However, it cannot operate all modes which are strictly reserved for channel founder. Channel operator is, however able to administrate the channel, set some modes on the channel, remove a badly behaving client from the channel, and promote other clients to become channel operator.

7.1 Channel Message Delivery

All clients that have joined the channel can send messages to the channel. All channel messages are secured and authenticated by channel key. The channel key is generated by the server when the channel is created, a client joins the channel, or a client leaves the channel. The channel key is also regenerated periodically. The reason for the regeneration of channel key everytime someone joins or leaves the channel is that it prevents new clients joining the channel, and old clients leaving the channel, to encrypt or decrypt old or new messages. They can encrypt and decrypt channel messages only when they have joined on the channel.

Channel keys are cell specific in the SILC Network. Each cell that have clients joined on a particular channel have also own key for the channel. That key is not shared by other cells in the network. Inside the cell the channel key is known by the router and all servers that have clients on the channel and all clients that have joined the channel.

Channel Message Delivery



The diagram above illustrates typical delivery of channel messages inside a cell and between two cells. Both of the cells have their own channel key. Both cells know all clients joined on the channel. When a message is sent to the channel by a client, it is encrypted with the current channel key in that cell. The servers and the router in the local cell then route the message to all local clients who have joined the channel. If the channel has clients that belong to other cells in the network, the router will route the channel message to that cell. When channel messages are sent between routers, they are first decrypted with the current channel key, and then re-encrypted with the session key shared between the two routers. The router who receives the channel message then decrypts it with the session and re-encrypts it with the current channel key in that cell. It then distributes the channel message to all clients on the channel. The clients who have joined the channel always know the current channel key and can decrypt all channel messages they receive. Note that normal servers in the SILC network never decrypt the channel messages even though they have the key. There is no reason for servers to decrypt the message. The router decrypts the message only when sending it between two routers.

This method of channel message delivery is the default way to send channel messages in the SILC Network. However, this is not a perfect solution in all circumstances. If the clients joined on a particular channel cannot trust, or do not want to trust the servers and routers in the SILC Network, they can consider the fact that servers and routers know the channel key is actually a breach of security.

If the clients on the other hand can trust their servers and routers in the SILC Network this is the recommended way of sending channel messages. This method is the simplest method for end user since it does not require any special settings before engaging the conversation on the channel. The client merely joins the channel, receives the channel key from the server and can start the conversation on the channel.

In addition of encrypting channel messages it also possible to digitally sign all sent channel messages. The receiver could then verify the signature of each of the message using the sender's public key.

7.2 Channel Message Delivery With Channel Private Key

If the clients cannot trust the servers and routers in the SILC Network they should not use the default way of sending the channel messages. Instead, they should use channel private keys to encrypt and decrypt the channel messages. Channel private keys are keys that are known only by the clients who have joined the channel. Servers and routers do not know the key and cannot decrypt the messages. When message is sent between two routers they are merely re-encrypted with the session key but not decrypted since the router do not have the key to do that.

The clients who have joined the channel must first agree on the channel private key they are going to use. The key may generally be anything. It may be a passphrase or a random string, or the key may negotiated using some key exchange protocol which provides negotiating the key for multiple clients at the same time.

As the channel private key is actually entirely local setting in the client, it is possible to set several channel private keys for one channel. It is possible to have multiple channel private keys that are not known by all channel members. When encrypting messages with one channel private key only the clients who have that key can decrypt the message. The other key could be shared for example by all clients on the channel and thus all clients can decrypt messages encrypted with that key. In this way it is actually possible to have a private group conversation inside the channel while having global conversation at the same time.

8 Private Messages

Private messages are messages that are sent from one client to another through the SILC Network. They are private because they are not sent to anyone else except to the true receiver of the message. Private messages can be used to engage private conversation with another client if channels are not desired.

As all messages in SILC the private message are also encrypted and authenticated. There are several ways to secure private messages. By default private messages are encrypted using the session keys established in the SKE protocol. It is also possible to negotiate a private message key between the two clients and encrypt the messages with that key. It is even possible to encrypt the messages with public key cryptosystem, if desired. The next sections will describe all these private message delivery methods.

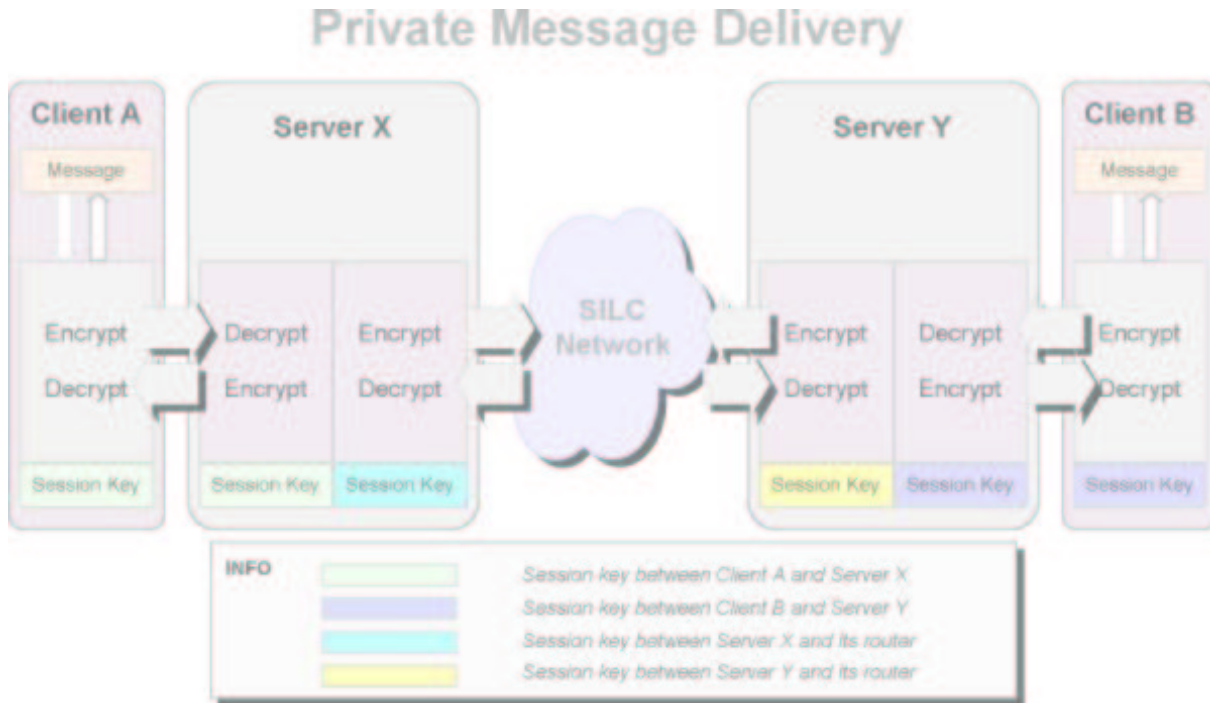
The SILC protocol provides these three methods of delivering private messages because none of the methods alone can satisfy the security requirements of all people. The end user should decide the acceptable level of risk, the required level of security and other security

and usability aspects when deciding what way of sending private message suites for them.

In addition of encrypting private messages it also possible to digitally sign all sent private messages. The receiver could then verify the signature of each of the message using the sender's public key.

8.1 Private Message Delivery With Session Keys

Sending private messages are by default secured with session keys established in the SKE protocol. This means that the private message is always encrypted with the session key of the next receiver of the message enroute to the receiving client. This also means that the message is decrypted and re-encrypted everytime it is sent further to the receiving client.



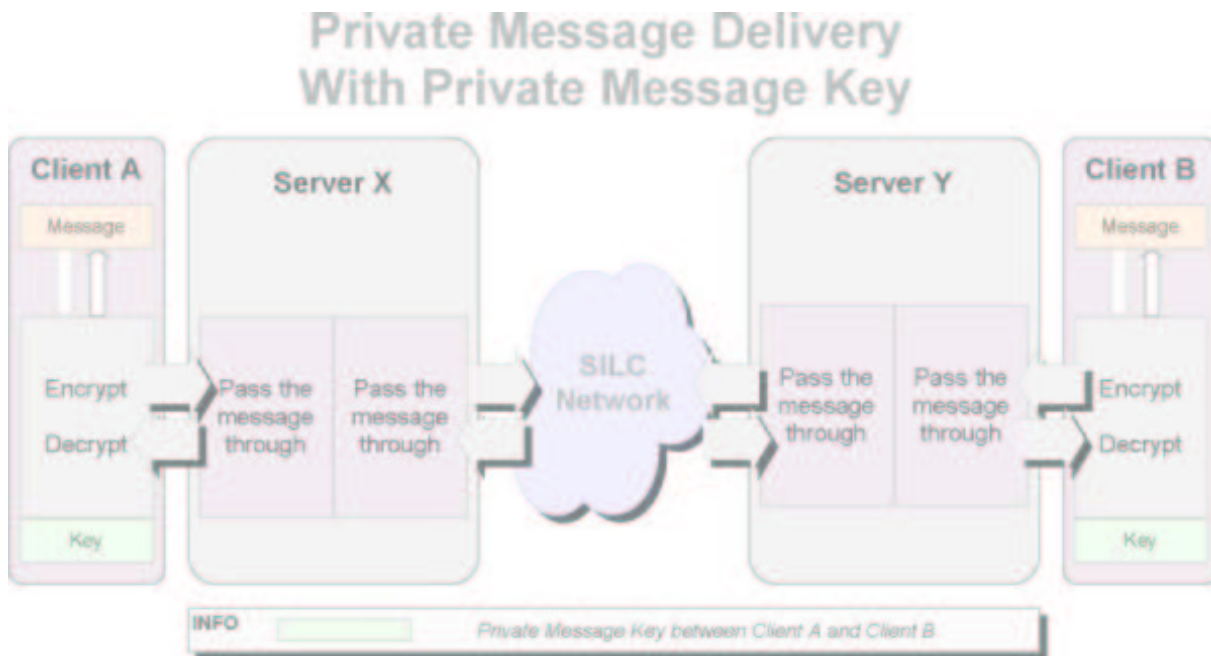
As the diagram above shows the private messages sent by Client A to the Client B travels through the SILC Network and is always decrypted and re-encrypted with the session key of the next receiver. The Client B then finally decrypts the private messages that is encrypted with the session key shared between the Client B and the Server Y.

This way of securing private messages is not perfect and cannot be used in all circumstances. If the clients having the conversation cannot trust the servers and routers in the SILC Network they should not send private messages that are secured in this manner. Messages secured in this manner can be decrypted by the servers and routers that the clients may consider to be untrusted.

If the clients on the other hand trust the servers and routers in their SILC Network, or they do not care that servers can decrypt their messages, sending private messages in this way is very simple from client's point of view. For servers and routers this of course means that they need to decrypt and re-encrypt each private message. Since this way of securing private message cannot be used at all times the SILC protocol provides other ways of securing private messages.

8.2 Private Message Delivery With Private Message Key

Private messages can be secured with private message key as well. This key is known only by the sender of the message and the receiver of the message. This way no one else except the sender and the receiver can encrypt and decrypt the private messages. The message is encrypted by the sender with the private message key and all the servers and routers pass the message through enroute to the receiver. They cannot decrypt the message since they do not have the key. When sending private messages in this way it does not matter whether the clients trust or do not trust the servers and routers in the SILC network.



As the diagram above shows the Client A encrypts the message with private message key and sends the message to the SILC Network. All servers and routers merely pass the message through since they cannot decrypt it. The Client B then receives the message and decrypts it with the private message key.

Sending private messages in this manner is always secure since the key is shared only by the sender and the receiver. The problem of this method is that the sender and the receiver must somehow agree about the key they are going to use. The private message key can generally be anything. It can be a passphrase that only the sender and the receiver knows. They could have been agreed to use some word or phrase as the key sometime earlier before they started the conversation. Or the key maybe from some random string from a code book that only the sender and the receiver poses. Or it can be a key that is negotiated using some key exchange protocol.

The problem however is fundamental. How to agree to use some key when you cannot reach the other person over secure channel? The SILC protocol solves this problem by providing a possibility to negotiate the key between two clients using the SKE protocol. One or both of the clients can set up the SKE server running in their host and ask the other client to connect to it. In this case the SKE is executed outside the SILC Network. As a result of the SKE protocol the clients have now shared secret that they can use as private message key. The key is known only by the two clients that executed the SKE protocol. They can then use that key to

secure all subsequent private messages.

Using this method of private messages delivery is recommended if the clients cannot trust the servers and routers in the SILC Network. The drawback is the extra phase of setting the private message key before starting the conversation. However, using the SKE protocol is the recommended way to negotiate the private message key since it can be automatized and does not cause any extra tasks for end user.

9 MIME Messages

SILC Protocol supports MIME messages as normal channel and private messages. By using MIME messages it is possible to send for example images, music and video and audio stream in SILC. Any MIME type that is supported by the application can be sent via SILC network.

The MIME messages are utilized by using so called Message Flags in the message payload that is used in SILC protocol. The Message Flags indicates the recipient that the message is a MIME message and it then knows how to interpret the message. Using Message Flags it possible also to send other kind of messages and to augment features of normal channel and private messages.

10 Secure File Transfers

The file transfer support in chat protocols are a absolute requirement nowadays, and chat protocol without one is no chat protocol at all. SILC also supports file transfer with the addition that the file transfer stream is secured. When a user wants to transfer a file to another user, the SILC Key Exchange (SKE) protocol is first executed to negotiate a session key for the file transfer stream. This key is then used to protect the peer to peer stream between users.

The file transfer protocol used in SILC protocol is the SSH File Transfer protocol (SFTP). Even though the name of the protocol relates to SSH, the actual file transfer protocol has nothing to do with Secure Shell. The SFTP is totally independent file transfer protocol and its stream is secured using SILC. The SFTP is very good protocol because in addition of providing simple file transfer support, it can also support complex file and directory manipulation.

The support for file transfer in SILC has been designed so that using practically any file transfer protocol is possible. The mandatory protocol is SFTP but in the future adding support for other protocols is also possible.

11 Future of the Protocol

The protocol has matured into the version 1.2 over the past few years. It has reached a level where it is the most rich featured conferencing protocol as of today. It is the SILC Project's intention to standardize the SILC protocol in the IETF and this is where the focus is now moving.

12 Conclusion

Secure Internet Live Conferencing is a modern conferencing protocol which provides rich conferencing features with high security. It has a wide range of security properties and features that should meet the highest levels of security requirements, while not forgetting ease of use. The network topology offers new architectural solution with better scalability over traditional chat protocols.

13 Further Information

More detailed information about the SILC protocol is available in the SILC protocol specification documents. There exists currently six Internet Drafts that defines the protocol in great detail. The Internet Drafts are available from the SILC Project website (<http://silcnet.org/>) but also from the IETF website (<http://www.ietf.org/>).

For comprehensive introduction to cryptography refer to the Cryptography A-2-Z document (<http://www.ssh.com/tech/crypto/>).

14 Terms and Abbreviations

- Asymmetric cryptosystem

Asymmetric cryptosystem provides public encryption. It has two keys, one public key and one private key (also called as secret key). The public key is publicly available allowing anyone to encrypt messages with the public key. Only the possessor of the private key can decrypt those messages. Difference to symmetric cryptosystem is that symmetric cryptosystem use only one key, and the key is usually used to both encryption and decryption. The asymmetric cryptosystem is also called as public key encryption, public key cryptosystem or public key algorithm. SILC supports RSA and DSS asymmetric cryptosystems.

- Authentication

The verification of the identity of a person, host or process in order to gain access to a service or prove identity. In data communications it also means verifying the origin of a message.

- Certificate

Certificate is a digital document which can be used to verify the identity of a person or host. In SILC, certificates can be used to prove identity of clients, servers and routers. Basically certificate is a public key with subject name. SILC supports X.509, OpenPGP and SPKI certificates. Supported public keys are SILC style public key and SSH2 style public key.

- Certification Authority (CA)

A third party entity that can verify identity of a person or host. CA is usually external company that provides certificates and their verification services.

- Diffie-Hellman key exchange

First public key algorithm ever invented. It is used to generate a secret key between two or more parties. It gets its security from the difficulty of calculating discrete logarithms.

- Encryption

A mechanism (usually mathematical) to transfer plaintext (or cleartext) to ciphertext to provide confidentiality. A process to transfer the ciphertext back to plaintext is called decryption.

- Integrity

The verification of data to detect any modifications. If data is modified enroute from the sender to the receiver, the modification will be detected.

- HMAC

Hash Message Authentication Code. Also called as keyed hash function. It is a secret key authentication algorithm which proves that the message is not modified and that the HMAC was computed by the sender of the message.

- Key management

Key management is a set of processes and mechanisms which support key exchange and maintenance of current keying relationships between parties, including replacing older keys with new keys as necessary, by executing rekey.

- Man-in-the-middle attack

An attack against two connecting entities where the attacker executes key exchange protocol with both of the parties independently without their knowledge. Both of the connecting entities will end up having secret key with the attacker, and the attacker can encrypt and decrypt all the messages that goes between the two entities.

- Message Authentication Code (MAC)

MAC provides message integrity by computing the MAC using a secret key authentication algorithm (HMAC).

- Perfect Forward Secrecy (PFS)

A property of rekey (or key regeneration) which defines whether the new key is derived from the old key. If Perfect Forward Secrecy is selected the new key is never dependent of the old key which means that if the old key would get compromised at later time it will not compromise the new key. In SILC setting PFS in the SKE protocol means executing the SKE protocol again. If PFS is not selected the new key is always derived from the old key.

- Rekey

A key regeneration process where the old key has expired or is not secure anymore to use. In this case rekey is performed and new key is generated.

- Symmetric cryptosystem

Symmetric cryptosystem is one key cryptosystem where one key is used usually to both encryption and decryption process. The symmetric cryptosystems are usually significantly faster than asymmetric cryptosystems. DES, AES, Twofish and Blowfish are examples of symmetric cryptosystems. SILC supports all the common symmetric cryptosystems including AES. SILC does not support DES as it is insecure and 3DES as it is too slow.