Mobility Extensions (MEXT)
Internet-Draft
Intended status: Standards Track
Expires: January 10, 2011

B. Patil
Nokia
D. Premec
Unaffiliated
C. Perkins
Tellabs
H. Tschofenig
Nokia Siemens Networks

July 11, 2010

Problems with the use of IPsec as the security protocol for Mobile IPv6

draft-patil-mext-mip6issueswithipsec-03

Status of this Memo

Copyright Notice

Abstract

Mobile IPv6 as specified in RFC3775 relies on IPsec for securing the signaling messages and user plane traffic between the mobile node and home agent. An IPsec SA between the mobile node and the home agent provides security for the mobility signaling. Use of IPsec for securing the data traffic between the mobile node and home agent is optional. This document analyses the implications of the design decision to mandate IPsec as the default security protocol for Mobile IPv6 and consequently Dual-stack Mobile IPv6 and recommends revisiting this decision in view of the experience gained from implementation and adoption in other standards bodies.

Table of Contents

1. Introduction

Mobile IPv6 as specified in RFC3775 [RFC3775] requires an IPsec security association between the mobile node (MN) and home agent (HA). The IPsec SA protects the mobility signaling messages between the MN and HA. The user data may be optionally protected by the IPsec SA but is not required. The use of IPsec by most hosts today is primarily as a solution for enterprise connectivity through VPN applications. IPsec has not evolved into a generic security protocol.

The use of IPsec and IKE (v1 and v2) with Mobile IPv6 are specified in RFCs 3776 [RFC3776] and 4877 [RFC4877]. The Mobile IP and MIP6 working groups in the IETF chose to mandate IPsec as the default security protocol for Mobile IPv6 based on various criteria and lengthy discussions that occured between the years 2000 and 2004. Implementation experience with Mobile IPv6 and the security variants with which it has been specified in some SDOs indicates a need to revisit the design choice for MIP6 signaling security. The analysis and recommendation to revisit the security protocol architecture for MIP6 should not be interpreted as a recommendation for Authentication Protocol for Mobile IPv6 [RFC4285]. The objective is to highlight the misfit of IPsec and IKEv2 as the security protocol for MIP6 and hence the need for considering alternatives. A simpler security architecture for securing the signaling and traffic between the MN and HA can co-exist with the IPsec based solution as well.

The objective of Mobile IPv6 [RFC3775] is to enable IP mobility for IPv6 hosts. The security aspect of the protocol is a critical component for consideration in terms of deployment and operation on large scales. If complexity of implementation were a consideration then the current specification dealing with Mobile IPv6, i.e RFC3775 and RFC5555 would win high accolades. An implementer spends 20% of his time on implementing the Mobile IPv6 protocol and 80% of the time integrating it with IPsec and IKEv2. And even after that interoperability of the client with home agents is not guaranteed. The IPsec/IKEv2 security architecture may work in implementations wherein the OS, the IPsec/IKEv2 stack and mobile ipv6 client software are all implemented by a single entity. It just does not work on open systems.

2. Terminology and Abbreviations

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

This document refers to [RFC3775][RFC4877] for terminology.

3. Background

IP mobility support in IPv6 was considered to be an integral feature of the IPv6 stack based on the experience gained from developing mobility support for IPv4. The design of Mobile IPv6 was worked on by the Mobile IP WG in the late 90s and by the MIP6 WG until its publication as [RFC3775] in 2004.

IPsec [RFC4301] was also intended to be a default component of the IPv6 stack and was the preferred protocol choice for use by any other IPv6 protocol that needed security. Rather than design security into

every protocol feature, the intent was to reuse a well-defined security protocol to meet the security needs. Hence Mobile IPv6 has been designed with a security architecture that relies on reusing IPsec.

The Mobile IPv6 specification [RFC3775] was published along with the companion specification "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", [RFC3776]. The establishment of the IPsec SA between the MN and HA as per RFC 3776 is based on the use of IKE. The use of IKE in the context of Mobile IPv6 for IPsec SA establishment did not gain traction because of factors such as complexity of IKE and the IETF transitioning to IKEv2. The MIP6 WG completed the specification, Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture [RFC4877] in April 2007. This [RFC4877] is considered as the default security protocol solution for MIP6 and updates [RFC3776].

4. Problem Statement

Mobile IPv6 is encumbered by its reliance on IPsec [RFC4301] from an implementation and deployment perspective. As a protocol solution for host based mobility, MIP6 can be simpler without the IPsec baggage. The issues with IPsec are even more exacerbated in the case of dual-stack MIP6 [RFC5555].

IPsec SAs between the MN and HA are established either manually or via the use of IKEv2 [RFC4306]. Manual SA configuration is not a scalable solution and hence MIP6 hosts and Home agents rely on IKEv2 for establishing dynamically IPsec SAs. As a result MIP6 depends on the existence of IPsec and IKEv2 for successful operation.

IPsec is unable to provide security protection for MIP6 in a transparent way, and numerous interactions between the host's security subsystems and the MIP6 application are needed in the course of the regular operation of the MIP6 application. Besides requiring an extensive communications channel between the security subsystems and the MIP6 application, those interactions often also require modification of the MNs security subsystems code. The situation today is such that the communications channel between the IPsec subsystems and the MIP6 application is non existent and this is generally true for most of the commercially available platforms. Even if such a channel were to be available, there does not exist a standardized protocol over that channel which would enable the MIP6 application to communicate with the security modules in a non-implementation specific way.

Considering a third party application developer who would like to

provide a MIP6 application for a particular platform, the need for numerous interactions with the IPsec subsystem and the unavailability of the standardized communications channel through which such interactions could take place is a major obstacle to the implementation of the mobility protocol. Without such a communication channel being available it is not possible to implement a MIP6 application as a third party developer.

Even if the platform would provide such a communication interface for the MIP6 daemon, this is still insufficient as the MIP6 protocol standardized today [RFC3775] requires numerous changes to the host's IPsec and IKEv2 implementation. This document enumerates various

implementation issues related to the interactions between the MIP6 application and the host's security subsystems.

An argument can be made that the MIP6 application itself should provide the required changes to the IPsec subsystems of the platform (maybe in the form of patches). While this is possible at least for some open source platforms to provide modifications to the host's IPsec implementation as well as the key management application(s), this is still an issue for several reasons:

- Target platform could be a commercial platform for which no source code for the security modules (IPsec and IKEv2) is available.
- If the MIP6 application were to patch the IPsec subsystems, then multiple MIP6 applications from different developers would implement it in different ways, which would inevitably lead to variations and problems with interoperability at a minimum, for instance when the user tries to install several MIP6 applications it is difficult to determine which one is the best suited for his/her needs.
- Key management daemons are usually provided as third party software for which no source code may be available, even if the platform itself is available as open source.
- Even if the MIP6 application developer would be willing to provide patches for the key management daemon to make it work with his MIP6 application, how would the MIP6 application developer know which of the several available key management daemons the user is running?
- Each application would be able to work only with a single key management daemon, namely the one for which the MIP6 application provides the patches. The user may be running another key management daemon and may be unwilling to change its daemon to the one that comes as part of the MIP6 application.
- Patches for the IPsec part in the kernel and the key management daemon would typically be valid only for the particular version of the kernel and the key management daemon for which they were written. This might prevent the user from upgrading the platform

or applying OS security patches that are provided as part of the regular platform maintenance since this would in all probability make the MIP6 application defunct.
- Modifying the security subsystems by a third party is a security issue and users are generally advised to refrain from allowing the

security subsystems to be modified in any way.

- The developer may not have the knowledge or the time to modify the platform's IKEv2 and IPsec subsystems, although it might be perfectly capable to deliver the MIP6 application itself.
- There could be copyright issues as well that would prevent modifications of the platform's security subsystems or the delivery of the modifications by the third party.
- Even if the MIP6 application developer is able to come up with the necessary patches for the security subsystem, it is not realistic to expect the prospective user of MIPv6 to first patch the kernel and the key management daemons before using the MIPv6 service.

The above discussion shows why it is unrealistic to expect that the MIP6 application could provide the needed modifications to the IKEv2 and IPsec subsystems of the host. Section 6 presents a more technical discussion of various implementation issues related to the interworking between the MIP6 application and the IPsec/key management modules.

The problem in a nutshell for MIP6 is the dependence on IPsec and IKEv2 for successful operation.

5. General issues with the use of IPsec for MIP6 security

This section captures several issues with the use of IPsec by MIP6.

(1) The design of Mobile IPv6 emphasized the reuse of IPv6 features such as IPsec. IPsec for IPv4 was a bolt-on solution. With the increasing need for security, IPv6 designers chose to incorporate IPsec as a default feature. There existed an assumption in the MIP6 working group that every IPv6 host would have IPsec capability as a standard feature. While this is true in many host implementations today, the existence of IPsec in every IPv6 stack is not a given. Hence a host which needs to implement Mobile IPv6 must ensure that IPsec and IKEv2 are also available. As a result of this dependence, MIP6 is no longer a standalone host-based mobility protocol. A good example of a host based mobility protocol that works as a self-sufficient module is Mobile IPv4 [RFC3344]. The security associated with MIP4 signaling is integrated into the protocol itself. MIP4 has been successfully deployed on a large scale in several networks.

(2) IPsec use in most hosts is generally for the purpose of VPN connectivity to enterprises. It has not evolved into a generic security protocol that can be used by Mobile IPv6 easily. While RFC4877 does specify the details which enable only the MIP6 signaling to be encapsulated with IPsec, the general method of IPsec usage has been such that all traffic between a host and the IPsec gateway is carried via the tunnel. Selective application of IPsec to protocols is not the norm. Use of IPsec with Mobile IPv6 requires configuration which in many cases is not easily achievable because of reasons such as enterprise environments preventing changes to IPsec policies.

(3) A MIP6 home agent is one end of the IPsec SA in a many-to-one relationship. A MIP6 HA may support a very large number of mobile nodes which could be in the hundreds of thousands to millions. The ability to terminate a large number of IPsec SAs (millions) requires signifiant hardware and platform capability. The cost issues of such an HA are detrimental and hence act as a barrier to deployment.

(4) The implementation complexity of Mobile IPv6 is greatly increased because of the interaction with IKEv2. The complexity of the protocol implementation is even more so in the case of Dual stack MIP6 [RFC5555] wherein NAT traversal scenarios are considered.

(5) IPsec and IKEv2 are not implemented or available by default in every IPv6 or dual stack host. Mobile IPv6 support on such devices is not an option. Many low-end cellular hosts have IP stacks. The need for IPsec and IKEv2 in these devices is not important whereas mobility support is needed in many cases. A simpler security protocol than the use of IPsec/IKEv2 would make MIP6 much more attractive to implement and deploy.

(6) [RFC4877] which specifies the use of IKEv2 and IPsec with Mobile IPv6 essentially results in a variant of IPsec which is specific to Mobile IP. Hence this results in added complexity to implementations.

(7) Mobile IPv6 needs to be capable of being deployed in situations where alternative security mechanisms are already well-understood by the network administration. It should be possible to enable Mobile IPv6 to work in situations where alternative security mechanisms already supply the necessary authentication and privacy.

(8) IPsec has been successfully applied to VPN and other infrastructure operations, but not for general end-to-end applications. Thus, the granularity for selectors was originally not at all sufficient for Mobile IPv6.

(9) The way that the IPsec code sits in the usual kernel, and the access mechanisms for the SA database, are not very convenient for use by straightforward implementations of Mobile IPv6. Unusual calling sequences and parameter passing seems to be required on many platforms.

(10) In certain environments the use of IPsec and IKEv2 for establishing the SA is considered as an overhead. Bandwidth constrained links such as cellular networks and air interfaces which are in the licensed spectrum tend to be optimized for user traffic. MIP6 signaling with the IPsec overhead and the IKEv2 messages are viewed negatively. It is more acceptable to have signaling without IPsec encapsulation.

The issues listed above can be speculatively attributed as some of the causes for MIP6 not being implemented widely.

6. Implementation Issues with IPsec

6.1. Triggering IKEv2 on the MN

When the MIP6 application is invoked on the MN, as part of the initialization steps it is expected to install the appropriate SPD entries for protecting the mobility management signaling. Creation of the SPD entry works fine assuming that the MN is statically preconfigured with the HoA information since the HoA address is needed to create the SPD entries. Once the SPD entries are created, the MIP6 application generates the BU message and sends it via the socket. Based on the previously installed SPD entry the IP stack detects that the BU message needs IPsec protection and since there is no appropriate IPsec SA available, the OS kernel triggers the key management daemon to establish the needed IPsec SA.

Things are not that straightforward when the HoA is assigned dynamically. MIP6 allows the MN to obtain the HoA dynamically during the establishment of the initial IPsec SA with the HA [RFC4877] and in this case the HoA is provided in the CONF IKEv2 payload. How is the key management daemon triggered to establish the IPsec SA with the HA in this case? Normally there should be an SPD entry in the SPD with the HoA address as part of the selector and the outgoing BU message would be matched against that entry and this would trigger the kernel to request the establishment of the IPsec SA. But the

MIP6 application is not able to install the appropriate SPD entry nor to generate the BU message since it doesn't have yet the HoA that is needed for this, the HoA becomes available only later as part of the IPsec SA establishment. So this is sort of a chicken and egg problem: the HoA is needed to trigger the establishment of the IPsec SAs, but the HoA is not available prior to the IPsec SA being

established.

The solution to this issue could be an out-of-band communication channel between the MIP6 application and the key management daemon through which the MIP6 application could request the establishment of the appropriate IPsec SA from the key management daemon without having to install the appropriate SPD entries and generate the BU message.

6.2. Instructing IKEv2 to ask for the MN HoA/prefix

In case of dynamic HoA assignment, the MIP6 application needs to instruct the key management daemon to request the HoA information from the HA. The MIP6 application must be able to tell whether it would like to get the complete address or only the prefix [RFC5026] from the HA, and also whether it would like to get the IPv4 HoA as part of the IKEv2 exchange. This requires an interface between the MIP6 application and the key management daemon.

6.3. Providing the MN prefix to the IKEv2 daemon

When the key management daemon on the HA side receives a request from the initiator to allocate the MIP6_HOME_PREFIX it needs to get the prefix from the MIP6 daemon running on the HA. Therefore there must be a communication channel between the key management daemon and the MIP6 application through which the key management daemon could request the HoA/prefix information. Further, when assigning the prefix, the MIP6 application needs to create state and save the assigned address information and associate it with the MN which created the IPsec SA. So this looks like at this point there is a need to create the BCE in a some type of a "larval" state as a place where to save this information on the HA side.

Request to assign an address (IPv6 and/or IPv4) via the CONF payload raises an additional concern, namely, how does the key management

daemon know that it needs to obtain the address from the MIP6 application? A generic key management daemon would by default have some other means to provide the addresses in the CONF payload without consulting the MIP6 application, so there must be some method to tell the key management daemon that it should request the addresses from the MIP6 application. The author is not aware of any such method being available currently.

6.4. Registering the MN's FQDN in DNS

[RFC4877] allows the HA to register the MN's FQDN in the DNS. In this case the MN must provide the FQDN in the IDi payload in the IKE_AUTH step of the IKEv2 exchange. Consequently, there must be

some interface between the key management daemon and the MIP6 application on the HA side through which the FQDN could be made available to the MIP6 application so that it can register the FQDN in DNS.

6.5. Providing the Home Network Prefix to the MIP6 application

When the key management daemon on the MN side obtains the home network prefix information from the HA, it needs to relay this information to the MIP6 application. This again requires a communication channel between the key management daemon and the MIP6 application.

6.6. SPD Entry for the HoA on the MN side

Once the MIP6 application on the MN obtains the HoA (either assigned via the CONF IKEv2 payload [RFC4877] or autoconfigured from the MIP6_HOME_PREFIX [RFC5026]), the appropriate SPD entries need to be created in the SPD. Some key management daemons may require that they have full control of the SPD. In such cases the MIP6 application should not create the SPD entries by itself as this might confuse the MIP6 daemon and cause inconsistent state. Instead, the MIP6 application would need to instruct the key management daemon to create the appropriate SPD entries. So depending on the expectations of the key management daemon, the MIP6 application should either instruct the key management daemon to create the SPD entries or the MIP6 application should create them by itself at this point.

If the policy requires protection of the data traffic the SPD entries for the data traffic would also need to be created at this point.

6.7. SPD Entry for the HoA on the HA side

The creation of the SPD entry on the HA side for protecting the MN's mobility signaling is similar to what is happening on the MN side and is described in the previous section. As soon as the HA assigns an HoA it may proceed with the creation of the appropriate SPD entry. The SPD entries for protecting the data traffic should also be created at this time.

However, the issue gets more complicated in the case where the HA provides the prefix to the MN and the MN autoconfigures the HoA. In this case neither the key management daemon nor the MIP6 application on the HA are aware of the MN's autoconfigured HoA so neither of them is in a position to install an appropriate SPD entry during (or immediately after) the IKEv2 exchange. Even worse, since the autoconfigured MN address is not known on the HA side it is not clear what is the contents of the TSi and TSr payloads in the final

IKE_AUTH message sent by the HA. It is unclear whether or not the SA for protecting the MN's mobility signaling gets established at all in such a situation.

6.8. The K bit

The K bit [RFC3775] requires an interface between the IPsec subsystem and the MIP6 application that is not available today, at least not in a standardized form. Such an interface that would enable the support for the K bit has been proposed before and additional information how it might look like is available in [I-D.sugimoto-mip6-pfkey-migrate] and [I-D.ebalard-mext-pfkey-enhanced-migrate]. However, those proposals were not standardized and as such there is no publicly available interface specification that could be used by the third party MIP6 application developers to invoke the key management daemon and IPsec kernel. Note also that the MIP6 application must have a detailed knowledge about the established IPsec SAs (complete SPD entries, old and new tunnel end points) in order to be able to indicate to the key management daemon which SAs needs to be updated, which is not in the spirit of the original IPsec intention to provide security to the applications in a transparent manner.

6.9. UDP encapsulation of DSMIP6 signaling

The DSMIP6 specification enables the MIP6 enabled MN to roam in IPv4 networks [RFC5555]. To cope with NATs the DSMIP6 specification introduces a UDP encapsulation feature for the MIP6 signaling messages as well as for the data traffic. The UDP encapsulation feature requires very tight coupling between the IPsec subsystems and the MIP6 application.

To send the BU message the MIP6 daemon first needs to generate the BU message and then hand it over to the IPsec subsystem which adds the transport mode ESP protection. Then in the next step the message must go back from the kernel to the MIP6 daemon in the user space which adds DSMIP6 UDP encapsulation and then the packet is finally sent out on the interface.

When the UDP encapsulated Binding Acknowledgment message is received on the MN side, it is first delivered to the MIP6 application which strips the UDP header and then somehow hands over the stripped message to the kernel's IPsec subsystem. The IPsec subsystems takes care of the transport mode ESP protecting the BU message and after removing the ESP header delivers the BU message back to the MIP6 application.

6.10. Transport mode IPsec SAs and NATs

In order to establish an IPsec SA in the case of DSMIP6 when the MN is behind a NAT, it is required to use transport mode SAs only. Implementation experience at least has shown that it is not easily done and the operation itself of establishing the IPsec SA is flaky at best.

7. Conclusion

Examples in Section 6 show that there is a need for an extensive communication between the MIP6 application and the IPsec subsystem on the host. Standardizing such communication channel and having it

available in a commercial OS implementations is not a realistic proposal in any practical time frame. On the technical side, this is due to the fact that the IPsec is usually provided as part of the OS kernel and it is always difficult to convince the OS vendor to change the kernel and in particular the security related subsystems. The other difficulty is that the key management is usually provided as the user space service and as such there are multiple key management daemons available. Convincing vendors of various key management daemons to provide a unified or standardized communication channel for the MIP6 application might prove equally difficult and is not a realistic option either. Besides the technical reasons, there are also other non-technical reasons of business or political nature why such proposals would be unrealistic.

Therefore this draft recommends that an alternate security framework be considered for MIP6. This alternate mechanism should be self contained so that it can be developed and delivered as part of the MIP6 application itself (from an implementation perspective analogous to the way web browsers handle security today). This would enable third party developers that have no access or are otherwise not in a position to change the IPsec code of the platform they are developing for to come up with a self contained and working MIP6 application. Such alternative security mechanisms would remove one of the major impediments, i.e the interactions with IPsec - why it is so difficult today to implement a working MIP6 application. This would foster the diversity of the MIP6 solutions and should therefore have beneficial effects on the availability of MIP6 solutions and promote the adoption of MIP6 in general.

In order to make the Mobile IPv6 protocol a solution that is easy to implement and available in even low-end devices, it is necessary to simplify the protocol. The design or the security architecture for MIP6 needs to be revisited and a solution that simplifies the implementability of the protocol considered. The implementation

experience shows that a working solution of Mobile IPv6 is possible to build. However it is not easily done.

The authors recommend that while Mobile IPv6 and Dual-stack Mobile IPv6 implementations can indeed use IPsec and IKEv2 for the security, it should also be possible to rely on an alternative security framework. One such alternative security solution is proposed in

'Security architecture for Mobile IPv6 using TLS' [I-D.korhonen-mext-mip6-altsec].

8. Security Considerations

This I-D discusses the security architecture of Mobile IPv6 which is based on IPsec. The dependency on IPsec for security of MIP6 signaling is a detriment to the protocol implementation and deployment. Hence the current security architecture needs to be reconsidered.

The experience gained over the last few years indicates that IPsec cannot necessarily be used as a generic solution for security. The design choice made for MIP6 in the initial stages no longer are valid and is hampering the implementation and use.

9. IANA Considerations

This document does not have any information which requires IANA review.

10. Acknowledgements

Jouni Korhonen would like to point out the importance of sustained supply of caffeine rich coffee when doing IETF work. Authors would also like to recognize Satyabrata Sahu, NK Garg, Sandeep Minocha and Harsh Verma for working on the implementation.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3775] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC3776] Arkko, J., Devarapalli, V., and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents", RFC 3776, June 2004.

[RFC4877] Devarapalli, V. and F. Dupont, "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture", RFC 4877, April 2007.

[RFC5026] Giaretta, G., Kempf, J., and V. Devarapalli, "Mobile IPv6 Bootstrapping in Split Scenario", RFC 5026, October 2007.

[RFC5555] Soliman, H., "Mobile IPv6 Support for Dual Stack Hosts and Routers", RFC 5555, June 2009.

11.2. Informative References

[I-D.ebalard-mext-pfkey-enhanced-migrate]
Ebalard, A. and S. Decugis, "PF_KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE", draft-ebalard-mext-pfkey-enhanced-migrate-00 (work in progress), August 2008.

[I-D.korhonen-mext-mip6-altsec]
Korhonen, J., "Security architecture for Mobile IPv6 using TLS", draft-korhonen-mext-mip6-altsec-05.txt (work in progress), July 2010

[I-D.sugimoto-mip6-pfkey-migrate]
Sugimoto, S., Dupont, F., and M. Nakamura, "PF_KEY Extension as an Interface between Mobile IPv6 and IPsec/IKE", draft-sugimoto-mip6-pfkey-migrate-04 (work in progress), December 2007.

[RFC3344] Perkins, C., "IP Mobility Support for IPv4", RFC 3344, August 2002.

[RFC4285] Patel, A., Leung, K., Khalil, M., Akhtar, H., and K. Chowdhury, "Authentication Protocol for Mobile IPv6", RFC 4285, January 2006.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

Authors' Addresses

Basavaraj Patil
Nokia
6021 Connection Drive
Irving, TX 75039
USA

Email: basavaraj.patil@nokia.com

Domagoj Premec
Unaffiliated
Heinzelova 70a
Zagreb, 10000
CROATIA

Phone:
Fax:
Email: domagoj.premec.ext@gmail.com

Charles Perkins
Tellabs
3590 N. 1st Street, Suite 300
San Jose, CA 95134
USA

Email: charles.perkina@tellabs.com

Hannes Tschofenig
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Phone: +358 (50) 4871445
Email: Hannes.Tschofenig@gmx.net
URI: http://www.tschofenig.priv.at