IKE Context Transfer in an IPv6 Mobility Environment

Fabien Allard
France Télécom R&D
38-40 rue du Général Leclerc
F-92794 Issy-Les-Moulineaux
fabien.allard@orange-ftgroup.com

Jean-Marie Bonnin
GET/ENST Bretagne
CS17607
F-35576 Cesson Sévigné
jm.bonnin@telecom-bretagne.eu

Julien Bournelle
France Télécom R&D
38-40 rue du Général Leclerc
F-92794 Issy-Les-Moulineaux
julien.bournelle@orange-ftgroup.com

Jean-Michel Combes
France Télécom R&D
38-40 rue du Général Leclerc
F-92794 Issy-Les-Moulineaux
jeanmichel.combes@orange-ftgroup.com

ABSTRACT

Internet Security is a major goal for both ISPs[1] and their customers but security provisioning has a cost in terms of bandwidth consumption and cryptographic material computation. In a mobility context this security must be set up from scratch after each handover and for each customer. The context transfer mechanism provides an efficient way to re-establish security parameters. This mechanism aims to transfer suitable information between equipments in order to reduce handover time. The benefits for an operator would be to maintain the same security level during and after a handover while keeping costs as lower as possible. In this paper, we use context transfer in order to transfer the IPsec and IKE contexts related to a mobile node from a previous security gateway to a new one. The first purpose of this paper is to define the IKEv2 context and to provide a solution for handling SPIs[2] collisions using MOBIKE. The second aim of this paper is to set out an implementation of the Context Transfer Protocol for IPsec/IKEv1 in an IPv6 mobility environment and to provide performance results of such an optimisation.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and protection; C.2.1 [**Network Architecture and Design**]: Network communications; C.4 [**Performance of Systems**]: Design studies

General Terms

Design, Performance, Security, Standardization

[1]Internet Service Providers
[2]Security Parameter Indexes

MobiArch'08, August 22, 2008, Seattle, Washington, USA.

Keywords

Context transfer, CXTP, network security, network mobility, IPsec, IKEv1, IKEv2, MOBIKE, SPI collision.

1. INTRODUCTION

Security provisioning is a major requirement in an all-IP-based network architecture providing multimedia services, especially for mobile users. Indeed, IP communications are more vulnerable to attacks when mobile nodes use wireless links which are generally more accessible than wired links for an attacker. In the case of network accesses protected by IPsec [9, 10], access is secured by an IPsec tunnel mode Security Association (SA) established between a client of the network and a security gateway. This SA normally needs to be established during the network access phase by running an IKE [5, 6] exchange between the two SA endpoints. However, the duration of this IKE exchange make it impractical to use when the node is mobile and frequently change its access gateway, as during a handover. In most case, it is expected that real time traffic will be impacted by the handover. In a near future, the growth of the number of mobile nodes will increase the number of IPsec SAs handled by security gateways. Furthermore, IKE protocols (v1 or v2) are quite computationally intensive because of the Diffie-Hellman key exchange or the number of EAP roundtrips. Thus, data processing charge in access network equipments will be more and more important.

In this paper, we propose a context transfer-based solution for transferring IPsec/IKE states between security gateways. The aims of the context transfer mechanism are to transfer the network states information relevant to a mobile node, and to follow it during its movements. Hence, as soon as the mobile node moves, the states must be restored in the new equipment. A network state, typically called a *context*, is a set of information installed by services on network equipments in charge of controlling the access. Such services are known as *context transfer candidate services* and examples include IEEE 802.11i, IPsec and AAA[3] protocols [4], QoS[4] policy, header compression, etc. Therefore, a context

[3]Authentication, Authorization and Accounting
[4]Quality of Service

transfer protocol can help in avoiding a complex and time consuming re-establishement of these services at the new location.
The purpose of this paper is to extend our works about the viability of the context transfer mechanism for IPsec/IKE in an IPv6 mobility environment. In a previous paper [1], we defined the IPsec and IKEv1 context and how they could be transferred using CXTP [11]. In this article, after an overview of IPsec and IKEv2, we define the IKEv2 context. By showing that collisions of SPIs can occur after an IKEv2 context transfer, we explain how these collisions can be solved by defining a new MOBIKE [3] extension. Then, we describe our testbed where IPsec/IKEv1 context transfer is currently being implemented and we show in details the implementation of CXTP for IKEv1. Finally, we provide some performance results of our implementation.

2. TERMINOLOGY AND NOTATION

In this paper, we will use the following notations, partially based on the IKEv2 specification [6]:

MN : Mobile Node.
AR : Access Router.
HA : Home Agent.
CoA : Care-of Address.
SA : Security Association.
HDR : IKEv2 header.
N : Notify message type.
SK{...} : indicates that payloads contained between brackets are encrypted and integrity protected.

3. OVERVIEW OF IPSEC AND IKEV2

In order to better understand the context of our works, we quickly present the IPsec protocol suite (RFC 4301 [10]) and the Internet Key Exchange version 2 (RFC 4306 [6]) designed by IETF.

IPsec is a security framework that operates at the network layer by extending the IP packet header. It provides interoperable, high quality, cryptographically based security for IPv4/IPv6. The security services offered by IPsec include access control, connectionless integrity, encryption and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and/or upper layer protocols. These objectives are met through the use of two traffic security protocols, the Authentication Header (AH [7]) and the Encapsulating Security Payload (ESP [8]), and through the use of cryptographic key management procedures and protocols like Internet Key Exchange (IKE [6]). IPsec defines a *Security Association* as its primitive whose purpose is the protection of IP packets. SAs can operate in *transport* mode, where the IPsec data field begins with upper level packet headers (usually TCP, UDP, or ICMP), or in *tunnel* mode, where the IPsec data field begins with an entirely new IP packet header. When two hosts share an IPsec SA, their Operating Systems maintain records in two databases:

Security Association Database: This database contains all parameters related to each SA and is consulted in order to know how to process each packet (in and out).

Security Policy Database: This database is established and maintained by a user, an administrator or an application and describes the security policy to apply to each packet.

A security association in SAD can be set up either manually or dynamically. However, the manual case is limited both in security and scalability. Dynamic management of the IPsec parameters, for example using IKEv2, is a scalable solution: peers do not need to know each other in advance and only security policy i.e. the SPD has to be configured on each of them.

The IKEv2 protocol mutually authenticates two peers - the *initiator* and the *responder* - in order to dynamically and securely establish IPsec SAs. It uses a secret information (e.g. a pre-shared key or a key provided by EAP) to efficiently establish IPsec ESP SAs and/or IPsec AH SAs in both transport and tunnel modes and negotiates a set of cryptographic algorithms to be used by the SAs to protect the traffic that they carry. It can be divided in two main phases. In the first phase, called IKE_SA_INIT, the two peers establish an IKE SA to protect subsequent messages. In the second phase, called IKE_AUTH, the two peers authenticate each other using the Peer Authentication Database (PAD) and start to configure the IPsec SAs.

The Peer Authentication Database identifies the peers that are authorized to communicate with the security gateway, specifies the protocol and method used to authenticate each peer, contains the authentication data for each peer and provides a link between IKEv2 and the SPD for the policy lookup.

If others IPsec SAs are needed, the peers use the CREATE_CHILD_SA exchange which relies on previous authenticated IKE SA.

Thus, we have two kind of contexts for a mobile node: the IPsec context and the IKE context. The IPsec context contains the data stored into the SAD and the SPD while the IKEv2 context contains the parameters negotiated by IKE and the data stored into the PAD. In this paper, the association of these two kinds of context will be called the IPsec/IKE context. The IPsec context was defined in [1]. Thus, in the next section we will define the IKEv2 context.

3.1 The IKEv2 context

The transfer of the IKEv2 context, in addition of the IPsec context, is necessary because it would be not possible to negociate new IPsec SAs after their expiration. The aim of this section is thus to isolate the IKEv2 parameters which have to be transferred between ARs in order to continue an IKEv2 session after the handover. Some of the parameters are negotiated during the different IKEv2 phases and the others are installed on the peers before the negotiations. The IKE_SA_INIT phase establishes the following parameters:

- *Initiator's SPI*[5] which identifies the initiator of the IKE SA,
- *Responder's SPI* which identifies the responder of the IKE SA,
- *Cryptographic algorithms* which are an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group, and a pseudo-random function (prf),

[5]Security Parameters Index

- *SKEYSEED* from which all keys are derived for that IKE SA,
- *Ni, Nr* which are the initiator and responder nonces,
- *Lifetime* of the IKE SA.

The initiator's SPI and responder's SPI identify a unique IKE SA. Other parameters of this phase define the cryptographic algorithms and keys to use for this IKE SA. Hence, all these parameters are needed by the nAR in order to refresh the IKE SAs after a handover.

The IKE_AUTH and CREATE_CHILD_SA phases establish the cryptographic algorithms and lifetime of the IPsec SAs. These parameters are also needed in order to allow the negotiation of new IPsec SAs after the context transfer.

Finally, the PAD is composed of the following parameters:

- *Identifier* which identifies the peer,
- *Authentication protocol*,
- *Authentication method*,
- *Pre-shared secret* or *X.509 certificate*,

They are needed by the IKE_AUTH phase in order to authenticate the peer, and thus are a part of the IKEv2 context. Therefore, the IKEv2 context is composed of the parameters established during the IKE_SA_INIT, IKE_AUTH and CREATE_CHILD_SA phases and the data from the PAD relevant to the MN.

3.2 Solution for the SPI collisions problem after an IPsec/IKEv2 context transfer

After a context transfer, some parameters of the IPsec/IKEv2 context may need to be updated on the MN and must be configured on the nAR: the IP addresses of the MN and the new AR and the SPIs. Indeed, for example if other MNs are connected to the nAR with allocated SPIs, transferred SPIs may collide with existing ones. In this section, we show how to update the IP addresses and we propose a solution for negotiating new SPIs after an IPsec/IKEv2 context transfer using MOBIKE [3]. MOBIKE allows the IP addresses of IPsec peers to be updated but not the SPIs. Hence, we propose a MOBIKE extension for carrying new generated SPIs and a solution for handling SPI collisions. The advantages of such a solution are to use an existing framework (i.e. IKEv2/MOBIKE) for handling SPI collisions after an IPsec context transfer rather than defining a complete new solution.

3.2.1 Solution description

Figure 1 depicts the solution overview. The term *initiator* means the party who originally initiated the first IKE SA i.e. the mobile node. Hence, the *responder* is the previous access router or the new one, depending on where the mobile node is connected. However, in the proposed solution, the *responder* initiates a MOBIKE exchange. The reasons are given in section 3.2.4.

The solution follows the principles defined in MOBIKE. Therefore, it is based on IKEv2 messages exchanges of INFORMATIONAL type containing a NOTIFY payload. Theses messages will be used to update IP addresses of the peers and to negotiate the SPIs.

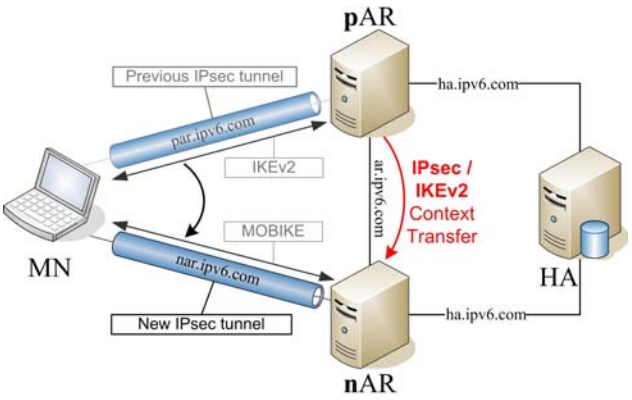

Figure 1: Architecture of the IPsec/IKEv2 context transfer

3.2.2 IP adresses update

MOBIKE allows to update IP addresses associated with an IPsec tunnel resulting from an IKEv2 exchange, i.e. an IPsec/IKEv2 context, when one or both of them change. For updating these addresses, exchanges defined in MOBIKE are integrally reused, in particular the UPDATE_SA_ADDRESSES type. They allow to configure the IP addresses for the transferred IPsec tunnel, which is built with the MN's previous CoA (pCoA) and the pAR IP address. As the responder is in charge of initiating the MOBIKE exchanges, it must know the nCoA in order to update the IPsec tunnel with the nCoA and the nAR IP address. The nCoA is learned through the Context Transfer Activate Request (CTAR) message sent by the mobile node to the new AR during the CXTP exchanges. We then get an updated IPsec tunnel built with the MN's new CoA (nCoA) and the nAR IP address (see figure 2).

3.2.3 Collisions of SPIs

A SPI is a value used to uniquely identify a security association. Two types of security associations exist:

IKE SAs used during the IKE exchanges to protect negotiations of IPsec SAs,

IPsec SAs used to protect the communications of IPsec peers.

Thus, there are SPIs for the IKE SAs and SPIs for IPsec SAs. An IKE SA is bi-directionnal while an IPsec SA is directional: On the one hand, an *inbound* SA is used when packets are received by a host and in the other hand an *outbound* SA is used when packets are sent by a host. We can thus note that a bi-directionnal communication (*inbound* and *outbound*) is characterized by two SPIs.

Depending on the selectors used for the SAD lookup, there are three different cases for the IPsec SAs SPIs:

Case 1 - SPI used alone for the SAD lookup
In this first case, each SPI must be unique in the system. It is thus possible to have a collision between the SPIs of the transferred IPsec SAs (*inbound* and *outbound*) and the SPIs of IPsec SAs currently in use in the new AR. Hence, the idea is to update the SPIs colliding in the new AR and to inform the MN by sending an IKEv2 notification message. The algorithm 1 is then followed.

Case 2 - SPI and destination IP address used for the SAD lookup
In this second case, a SPI collisions may occur when the IP address destination is the router's one (*inbound* SAs). As for the previous case, the idea is to update the colliding SPIs in the router and the MN by sending an IKEv2 notification message.

Case 3 - SPI, destination IP address and source IP address used for the SAD lookup
In this last case, there is no risk of collisions, since the new MN's CoA is necessarily in the lookup triplet and is not used in the new AR's SAD.

Algorithm 1 SPI collisions process algorithm

```
set MAX_ROUND_TRIPS
set TIMER
rt ← 1
while (SPI collision on nAR) do
  send N(UPDATE_SPI) to MN
  if (rt > MAX_ROUND_TRIPS) or (TIMER expires) then
    relaunch a complete IKEv2 negociation
  else
    rt ← rt + 1
  end if
end while
```

Now, regarding the IKE SAs SPIs, a collision can occur for each case defined previously since an IKE SA is only selected by the initiator and responder SPIs couple.

3.2.4 *Messages exchanges for SPIs negotiation*

In MOBIKE, the initiator decides which addresses are used in the IPsec SAs. That is, the responder does not normally update any IPsec SA without receiving an explicit UPDATE_SA_ADDRESSES request from the initiator. However, if a SPI collision occurs in the new AR, when the mobile node will initiate a MOBIKE exchange, the new AR will not be able to handle the request. Therefore, in our solution the responder is in charge of initiating a MOBIKE exchange. This is allowed (see MOBIKE [3] section 3.5) only when the source address that the responder is currently using becomes unavailable. We are typically in this case since after an IKEv2/IPsec context transfer, the IP address of the previous AR becomes unavailable in the new AR.

Since MOBIKE does not allow to update the SPIs, we define a new INFORMATIONAL request containing the UPDATE_SPI notification. We also define the NOTIFICATION_DATA payload for this new notification type:

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                                                               !
~              New Security Parameter Index (SPI)               ~
!                                                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!D!                       Padding                               !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   o New SPI (variable length) - New generated SPI in order to
     avoid the collision,
   o D flag (1 bit) - Direction flag used to know if the SPI has
     to be modified for the initiator (0) or the responder (1).
```

Therefore, if a new SPI must be negotiated, the NOTIFY payload fields will be the following:

```
   o PROTOCOL_ID = 1 if there is for an IKE SA, 2 (AH) or
                   3 (ESP) if there is for an IPsec SA.
   o NOTIFY_MESSAGE_TYPE = UPDATE_SPI
   o SPI         = SPI which has to be modified.
   o NOTIFICATION_DATA containing:
     - New SPI   = New generated SPI.
     - D         = 0 or 1
```

The UPDATE_SPI notification can be sent in the same request as the UPDATE_SA_ADDRESSES notification. If several SPIs need to be updated, several UPDATE_SPI notifications containing each one a new SPI can be inserted in the same request.

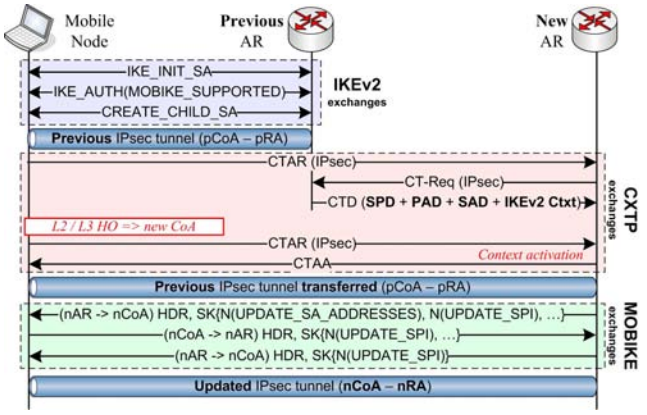

Figure 2: Overview of messages exchanges for the IPsec/IKEv2 context transfer and the context configuration using our solution

At the end of the MOBIKE exchanges, the IPsec databases i.e. SAD, SPD and PAD and also the IKEv2 context are modified with the new negotiated parameters. The IPsec tunnel is thus updated and network services depending on the security establishment can continue.

In the next section, the implementation of IPsec/IKEv1 context transfer using CXTP will be described and evaluation results will be provided. However these results do not take into account the case of SPI collisions because the SPI collision handling part is not implemented yet.

4. IMPLEMENTATION OF THE IPSEC / IKE CONTEXT TRANSFER IN AN IPV6 MOBILITY ENVIRONMENT

The context of our works is the following: a mobile node using Mobile IPv6 sets up a an IPsec tunnel with an access router after a successfull authentication. Then, an IKE exchange is performed between the MN and the AR in order to configure their IPsec databases. When a handover occurs, the MN should re-authenticate itself to regain access to the network since the new AR IPsec databases are not configured. Thus, the whole authentication processing has to be set up from the beginning to re-establish the IPsec tunnel. In this section we propose an implementation of the context transfer for IPsec/IKEv1 using CXTP and we provide some results obtained from this implementation.

4.1 Tesbed description and assumptions

Our platform (see figure 1) is made of four stations running FreeBSD 5.4: 1 MN, 2 ARs and 1 HA. Both pAR and

	Average delay (in ms)	Number of messages	Total size of messages (in Bytes)
IKEv1 main mode	1500	11	2182
IKEv1 aggressive mode	1300	8	1896
IKEv1 with context transfer optimisation	20	1 for context activation [1 for acknowledgment]	106 [82]

Table 1: Comparison results between IKEv1 with and without context transfer optimisation for setting up an IPsec tunnel after a handover.

nAR are connected to the Internet and to the HA. Initially, the MN is connected to the pAR and his traffic is protected by an IPsec tunnel using ESP. We use a KAME[6] snap to support Mobile IPv6. We use Racoon[7] instead of Racoon2 as IKE daemon and therefore we transfer IKEv1 context. This is because the works on IPsec context transfer initially began when IKEv2 daemon was not available. We can note that in IKEv1, there is no PAD, thus the IPsec/IKEv1 is composed by SAD's data, SPD's data and informations obtained from IKE phase 1. As we use IKEv1 instead of IKEv2, we can not use MOBIKE to configure the IP adresses of the IPsec tunnel. We thus have implemented a solution which automatically update the contexts with the new IP adresses.

The network address plan is the following:

- MN's CoA: `MN.par.ipv6.com`
- pAR's IP@: `pAR.par.ipv6.com`
- nAR's IP@: `nAR.nar.ipv6.com`
- HA's IP@: `HA.ha.ipv6.com`

Before the handover, the MN's SPD is configured with the following parameters:

```
#### SPD configuration ####
spdadd MN.par.ipv6.com HA.ha.ipv6.com ipv6 -P out ipsec
esp/tunnel/MN.par.ipv6.com-pAR.par.ipv6.com/require;

spdadd HA.ha.ipv6.com MN.par.ipv6.com ipv6 -P in ipsec
esp/tunnel/MN.par.ipv6.com-pAR.par.ipv6.com/require;
```

The SPD configuration file indicates that a tunnel protected by ESP is required between MN (`MN.par.ipv6.com`) and pAR (`pAR.par.ipv6.com`) in both direction (*in* and *out*) for the MN-HA traffic. The pAR's SPD is similar to the MN's one but directions (i.e. `in` and `out`) are switched. By default, the nAR's SPD discard the IP traffic from any MN.

4.2 Implementation modules

The implementation (see figure 3) is divided into two modules. A first one, named *CXTP module* which follows the guidelines from RFC 4067 (CXTP) and a second one, named *IPsec CXTP module* which links the CXTP module with the FreeBSD kernel's IPsec databases and Racoon. These modules intercommunicate through a shared memory where the contexts are stored and by using signals. This is done to guarantee that CXTP can work with every kind of context, not only with IPsec context.

[6]http://www.kame.net
[7]http://ipsec-tools.sourceforge.net

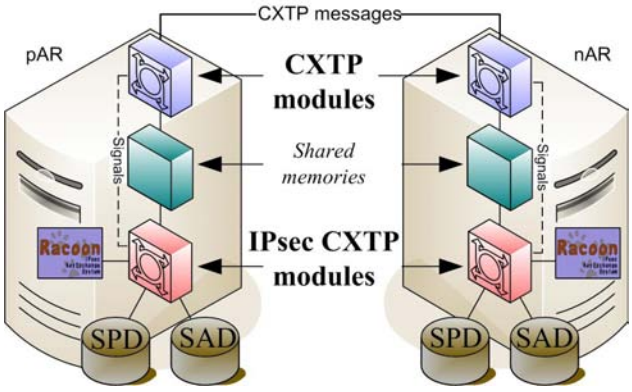

Figure 3: Implementation architecture

4.3 Evaluation results

In order to get the following results, we used an UDP traffic generator with a delay of 50 ms between each packet. We do not care about the handover delay, i.e. the delay took by mobile IPv6 to re-establish the MN-HA tunnel. We are only focused on the security set up delay, i.e. the delay took by IKE with and without context transfer optimisation to set up the IPsec tunnel between the MN and the nAR. During the time, all UDP packets are lost. After that delay, UDP traffic is thus allowed to reach his destination. Therefore, we have two main cases:

1. IKEv1 without optimisation: In this case, the MN performs an IKEv1 exchange with the pAR and then moves to the nAR. At this moment, it gets a new CoA from the nAR and relaunches the whole IKEv1 process in order to regain access to the Internet. As IKEv1 provides two modes for the phase 1 (i.e. *main* mode and *agressive* mode used to reduce round trips), we tested both of them.

2. IKEv1 with context transfer optimisation: In this case, the MN performs an IKEv1 exchange with the pAR and performs a predictive context transfer in order to transfer the IPsec/IKEv1 context to the nAR. Then, it moves to the nAR and gets a new CoA. At this moment, it activates the context by sending a CTAR message which may be acknowledged and regains access to the Internet. Then, it can relaunch the whole IKEv1 process in order to refresh the keys.

For the IKEv1 without optimisation case, results take into account the delay of network exchanges and delay of crypto-

graphic material computation. For the IKEv1 with context transfer optimisation, results take only into account the delay of network exchanges since there is no computation of cryptographic material.

Impacts on the UDP traffic can be show figure 4. Differents delays can be shown:

α is the handover delay,

β is the delay of the IPsec tunnel re-establishment in case of IKEv1 is used **with context transfer optimisation**.

γ is the delay of the IPsec tunnel re-establishment in case of IKEv1 is used in **aggressive mode**.

δ is the delay of the IPsec tunnel re-establishment in case of IKEv1 is used in **main mode**.

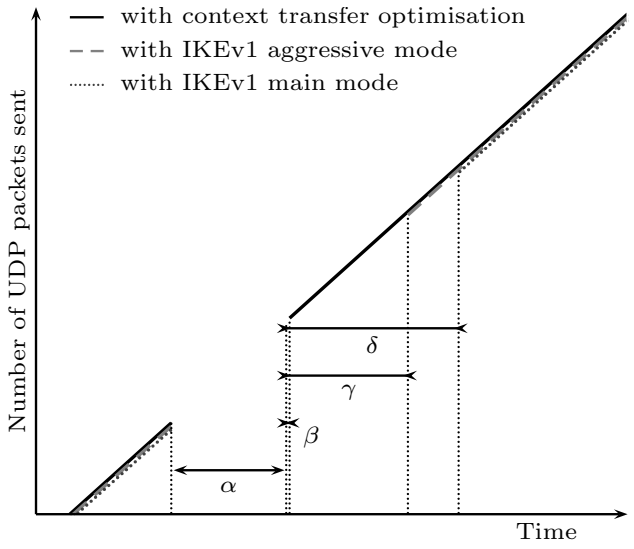

Figure 4: Impacts on UDP traffic

As we can see in table 1, with the IPsec/IKEv1 context transfer optimisation, the security set up takes only 20 ms while without this optimisation, it takes at least 1300 ms. In terms of number of messages exchanged, context transfer needs only one message in order to activate the context. Regarding IKEv1, 8 messages are needed at least to set up security parameters. This impacts the amount of data sent over the access link which is the most critical wire between the MN and the Internet.

5. CONCLUSION & FUTURE WORK

In this paper we set out a possible application of the context transfer mechanism for IKE. This mechanism can offer performance improvements for IPv6 mobility environment while guaranteeing an unchanged security level. After defining the IKEv2 context, we outlined a method for updating IP addresses in the IPsec/IKEv2 context and provided a solution for handling SPI collisions after a context transfer using MOBIKE. Then, we presented an implementation of CXTP for IPsec/IKEv1. We implemented a module to transfer generic contexts and another module to handle the IPsec SAs and the IKE SAs. Finally we provided comparison results between IKEv1 with and without context transfer optimisation for setting up an IPsec tunnel after a handover. Our next step will be to simulate this mechanism to measure performances benefits during handovers but also in order to compare it with other solutions such as pre-authentication [2]. These studies could help to apply the context transfer mechanism to issues like - IPsec Failover [12] - or - Home Agent Reliability [13] - both currently under study at IETF. For this purpose, we plan to submit an IETF draft about the proposed solutions.

6. REFERENCES

[1] F. Allard and J.-M. Bonnin. An application of the context transfer protocol: IPsec in a IPv6 mobility environment. In *Int. J. Communication Networks and Distributed Systems*, volume 1, pages 110–126. Inderscience Enterprises Ltd., February 2008.

[2] A. Dutta, V. Fajardo, Y. Ohba, K. Taniuchi, and H. Schulzrinne. A Framework of Media-Independent Pre-Authentication (MPA). Internet Draft, Internet Engineering Task Force, March 2007.

[3] P. Eronen. IKEv2 Mobility and Multihoming Protocol (MOBIKE). RFC 4555, Internet Engineering Task Force, June 2006.

[4] M. Georgiades, N. Akhtar, C. Politis, and R. Tafazolli. Enhancing mobility management protocols to minimise AAA impact on handoff performance. In *Computer Communications*, volume 30, pages 608–618. Butterworth-Heinemann, February 2007.

[5] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, Internet Engineering Task Force, November 1998.

[6] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. RFC 4306, Internet Engineering Task Force, December 2005.

[7] S. Kent. IP Authentication Header (AH). RFC 4302, Internet Engineering Task Force, December 2005.

[8] S. Kent. IP Encapsulating Security Payload (ESP). RFC 4303, Internet Engineering Task Force, December 2005.

[9] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, Internet Engineering Task Force, November 1998.

[10] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301, Internet Engineering Task Force, December 2005.

[11] J. Loughney, M. Nakhjiri, C. Perkins, and R. Koodli. Context Transfer Protocol (CXTP). RFC 4067 (Experimental), Internet Engineering Task Force, July 2005.

[12] V. Narayanan. IPsec Gateway Failover and Redundancy - Problem Statement and Goals. Internet Draft, Internet Engineering Task Force, December 2006.

[13] R. Wakikawa. Home Agent Reliability Protocol. Internet Draft, Internet Engineering Task Force, March 2007.